Mining Social Media Communities and Content

by
Akshay Java

Dissertation submitted to the Faculty of the Graduate School
of the University of Maryland in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2008

ABSTRACT

Title of Dissertation:	Mining Social Media Communities and Content
	Akshay Java, Doctor of Philosophy, 2008
Dissertation directed by:	Dr. Timothy W. Finin Professor Department of Computer Science and Electrical Engineering

Social Media is changing the way people find information, share knowledge and communicate with each other. The important factor contributing to the growth of these technologies is the ability to easily produce "user-generated content". Blogs, Twitter, Wikipedia, Flickr and YouTube are just a few examples of Web 2.0 tools that are drastically changing the Internet landscape today. These platforms allow users to produce and annotate content and more importantly, empower them to share information with their social network. Friends can in turn, comment and interact with the producer of the original content and also with each other. Such social interactions foster communities in online, social media systems. User-generated content and the social graph are thus the two essential elements of any social media system.

Given the vast amount of user-generated content being produced each day and the easy access to the social graph, *how can we analyze the structure and content of social media data to understand the nature of online communication and collaboration in social applications?* This thesis presents a systematic study of the social media landscape through the combined analysis of its special properties, structure and content.

First, we have developed a framework for analyzing social media content effectively. The BlogVox opinion retrieval system is a large scale blog indexing and content analysis engine. For a given query term, the system retrieves and ranks blog posts expressing sentiments (either positive or negative) towards the query terms. Further, we have developed a framework to index and *semantically* analyze syndicated[1] feeds from news websites. We use a sophisticated natural language processing system, OntoSem [163], to semantically

[1] RSS/ATOM

analyze news stories and build a rich fact repository of knowledge extracted from real-time feeds. It enables other applications to benefit from such deep semantic analysis by exporting the text meaning representations in Semantic Web language, OWL.

Secondly, we describe novel algorithms that utilize the special structure and properties of social graphs to detect communities in social media. Communities are an essential element of social media systems and detecting their structure and membership is critical in several real-world applications. Many algorithms for community detection are computationally expensive and generally, do not scale well for large networks. In this work we present an approach that benefits from the scale-free distribution of node degrees to extract communities efficiently. Social media sites frequently allow users to provide additional meta-data about the shared resources, usually in the form of tags or *folksonomies*. We have developed a new community detection algorithm that can combine information from tags and the structural information obtained from the graphs to effectively detect communities. We demonstrate how structure and content analysis in social media can benefit from the availability of rich meta-data and special properties.

Finally, we study social media systems from the user perspective. In the first study we present an analysis of how a large population of users subscribes and organizes the blog feeds that they read. This study has revealed interesting properties and characteristics of the way we consume information. We are the first to present an approach to what is now known as the *"feed distillation"* task, which involves finding relevant feeds for a given query term. Based on our understanding of feed subscription patterns we have built a prototype system that provides recommendations for new feeds to subscribe and measures the readership-based influence of blogs in different topics.

We are also the first to measure the usage and nature of communities in a relatively new phenomena called Microblogging. Microblogging is a new form of communication in which users can describe their current status in short posts distributed by instant messages, mobile phones, email or the Web. In this study, we present our observations of the microblogging phenomena and user intentions by studying the content, topological and geographical properties of such communities. We find that microblogging provides users with a more immediate form of communication to talk about their daily activities and to seek or share information.

The course of this research has highlighted several challenges that processing social media data presents. This class of problems requires us to re-think our approach to text mining, community and graph analysis. Comprehensive understanding of social media systems allows us to validate theories from social sciences and psychology, but on a scale much larger than ever imagined. Ultimately this leads to a better understanding of

how we communicate and interact with each other today and in future.

Mining Social Media Communities and Content

by
Akshay Java

Dissertation submitted to the Faculty of the Graduate School
of the University of Maryland in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2008

To my grandparents

ACKNOWLEDGEMENTS

Coming back to grad school to finish my Ph.D. after completing my Masters degree was one of the most important decisions I have made and I am glad that I did. The years spent at the University of Maryland, Baltimore County have taught me as much about life as it has about research. It is where I have made long lasting friendships and collaborated with individuals who have inspired me, challenged me and encouraged me. I am truly indebted to this insitituition.

My advisor, Prof. Tim Finin, has been my inspiration. Dr. Finin has always given me full support and allowed me to explore new topics and research problems that drove my interests. I am thankful for his words of advice and many skills I have gained by working with him.

Prof. Anupam Joshi has always kept me focused and made sure that I was making progress towards my dissertation. I am grateful to have Dr. Tim Oates, Dr. Charles Nicholas, Dr. James Mayfield on my committee. They have always made themselves available and accessible and I thank them for their time, suggestions and important advice.

It has been a privilege working with Prof. Sergei Nirenburg and ILIT lab members. Dr. Nirenburg's long term vision and persistence at solving one of the most difficult problems in computer science is inspirational.

All the ebiquity group members have been extremely supportive in building an atmosphere conducive to research. It has been a great joy working with friends, co-authors and colleagues Pranam Kolari and Anand Patwardhan. We have spent long hours discussing research, startups and life and I know I always have someone to turn to when I want to discuss some crazy idea.

I have been fortunate to have had the opportunity to collaborate with several people throughout my graduate studies. I am thankful to Palo Alto Research Center and Adam Rosien where I spent my first summer internship. I am grateful to my friend and mentor Dr. Eric Glover whith whom I had the opportunity to work at NEC Laboratories America Inc. Special thanks to Dr. Belle Tseng for her guidance and encouragement both during my second internship at NEC Laboratories America and supporting me throughout my graduate

career. I also thank Dr. Tseng for her time and guidance and for being a part of my dissertation committee. Thanks to Dr. Xiodan Song and Dr. Shenghuo Zhu for their advice and patiently answering my many questions. I have learned a great deal during these discussions, which have played an important role in my research.

A special note of thanks to the office staff of the Computer Science and Electrical Engineering department at UMBC, particularly Jane Gethmann. It would have been a lot harder trying to graduate on time without all the help from the wonderful staff here.

I thank my friends Medha Umarji, Priyang Rathod, Vishal Shanbhag and Mithun Sheshagiri, for all the good times and for always picking up my tab while I was still in school.

But above all, it is only due to the love, dedication, sacrifice and support of my family that I am here today. My grandparents, who have been a great influence in my life, have instilled in me the value and importance of learning throughout my life. My parents, who despite all the struggles in life, ensured that I was able to receive the best education. And all my family and friends who supported my graduate education both financially and by believing in me throughout. It is because of them that I am at this juncture today. Thank you all for being a part of my life.

TABLE OF CONTENTS

I. Introduction **1**

A. Thesis Statement 2

B. Thesis Outline 2

C. Thesis Contributions 5

II. Background and Related Work **7**

A. The Social Web 7

1. The Blogosphere 8

2. Social Networking Sites 9

3. Tagging and Folksonomies 11

B. Mining the Social Web 12

1. Structure and Properties 12

2. Mining Social Media Content 14

3. Communities in Social Graph 16

III. Mining Social Media Content **19**

A. Mining Sentiments and Opinions 20

1. Related Work 21

2. The TREC Blog Track 22

3. BlogVox Opinion Retrieval System 22

4. Data Cleaning 23

5. Evaluations 29

6. Conclusions . . . 40
B. Semantic Analysis of RSS Feeds . . . 42
1. Related Work . . . 42
2. OntoSem . . . 45
3. Making RSS Machine Readable . . . 50
4. Semantic News Framework . . . 57

IV. Mining Social Media Structure **65**
A. Approximate Community Detection in Social Media . . . 65
1. Related Work . . . 67
2. Sampling Based Low Rank Approximations . . . 69
3. Heuristic Method . . . 72
4. Evaluation . . . 72
5. Conclusions . . . 77
B. Simultaneous Clustering of Graphs and Folksonomies . . . 78
1. Related Work . . . 79
2. Clustering of Graph and Tags . . . 80
3. Dataset Description . . . 81
4. Evaluation . . . 82
5. Discussion . . . 88
6. Conclusions . . . 90
C. Microblogging Communities and Usage . . . 91
1. Dataset Description . . . 94
2. Microblogging Usage . . . 96
3. Mining User Intention . . . 104
4. Conclusions . . . 112

V. Influence and Trust **115**
A. Finding High Quality Feeds . . . 116
1. Related Work . . . 117
2. Dataset Description . . . 119

3. Clustering Related Topics 122
4. Applications 128
5. Evaluation 129
6. Conclusions 130
B. Epidemic Based Influence Models 132
1. Related Work 132
2. Cascade Models 132
3. Evaluation 135
4. Identifying Leaders using Influence Propagation 136
5. Conclusions 140

VI. Conclusions **141**

A. Discussion 143
B. Future Work and Open Problems 144

Chapter I.

INTRODUCTION

Social media is described as

> *an umbrella term that defines the various activities that integrate technology, social interaction, and the construction of words, pictures, videos and audio. This interaction, and the manner in which information is presented, depends on the varied perspectives and "building" of shared meaning, as people share their stories, and understandings.*[1]

Social Media has radically changed the way we communicate and share information both within and outside our social networks. The radical shift on the Web, from what typically was a one way communication, to a conversation style interaction has led to exciting new possibilities. Earlier, when a user posted her pictures from a recent vacation, there was little opportunity for her friends to comment on the photographs. Now, using sites like Flickr, friends can immediately see the uploaded pictures and post comments in response. The photographs are organized by means of albums and through the use of free-form descriptive tags (also known as folksonomies), which make them more findable. Moreover users can post their photos in communities which are often organized around different themes (like pictures of birds, locations, still photography, black and white photos etc). Such communities help foster an environment of sharing and allow users to share tips and receive feedback on their photography skills. A similar communication shift is occurring across media formats as new social media sites allow sharing text, photos, music, videos, podcasts and even PDF documents.

The key to the success of sites like YouTube, del.icio.us and Flickr is the underlying "Social Graph". Individuals can discover and post information and share content with their contacts in the social graph. A

[1] Wikipedia 08

social graph can be described as the sum of all declared social relationships across the participants in a given network. Studying the structure and properties of such graphs can be crucial in helping us understand the nature of online communication and perhaps even explain the success of social media.

The participatory nature of social media makes it different from the Web. Content produced in social media is often referred to as "user-generated content". As opposed to professionally edited text (news sites and magazine articles for example), user-generated content contributes to about five times more content present on the Web today. With almost 8-10 Gigs of data being produced each day by social media sites [171], many interesting questions arise on how we can analyze such content and study its utility? How do users participate and interact in such networks? What is the structure of such networks? How do individual interactions lead to community formation and what are the techniques to detect them efficiently?

The motivating question that has guided this thesis is the following: *"How can we analyze the structure and content of social media data to understand the nature of online communication and collaboration in social applications?"*.

A. Thesis Statement

It is possible to develop effective algorithms to detect Web-scale communities using their inherent properties structure and content.

This thesis is based on two key observations

- Understanding communication in social media requires identifying and modeling communities.
- Communities are a result of collective, social interactions and usage.

B. Thesis Outline

As part of this research, we have explored a wide range of social media platforms and graphs. RSS[2] and ATOM[3] formats have made it possible to share content in a timely and easily manner. The popularity of these XML based syndication mechanism ensured that blogs could now be read and rendered in a feed reader or a browser. It is no longer necessary to visit each site individually to check if there are any new updates. Many news sites and portals have now started offering RSS/ATOM feeds to their users. Indexing and processing

[2] Really Simple Syndication (RSS 2.0)
[3] Atom Publishing Protocol

these feeds meant that new applications could be built to have access to fresh and timely information. We started this research by building a natural language understanding agent to process RSS/ATOM feeds from news sites [85]. The news stories were processed using a sophisticated ontology based NLP system, OntoSem [163]. The motivation behind this work was to create large scale 'Fact Repositories' that would store the most current and up-to-date information about various events in news. Each news article was syntactically and semantically analyzed and the processed meaning representation was stored in a fact repository using the Semantic Web language OWL.

Our initial success in processing news motivated us to look into other social media datasets like blogs and wikis. This led to many interesting new challenges. While data obtained from news sites like BBC[4] and CNN[5] are usually excerpts from edited articles, blogs generally tend to be noisy and somewhat unstructured. We soon realized that processing blogs and social media data required new techniques to be developed. One of the main problems while dealing with blogs was that of spam. Jointly with Kolari et al. [110] we were the first to identify and address the problem of spam blogs in social media. We explored the use of new and innovative feature sets in a machine learning setting to identify and eliminate spam in the blogosphere. The ability to remove spam provided us an important advantage when developing future applications like opinion retrieval and community detection.

Blogs empower users with a channel to freely express themselves. Often this leads to a wide variety of content production online. Topics may range from popular themes like technology, politics to niche interests like kniting manga anime or obscure 60s LP music albums. More importantly, blogs provide a channel to discuss niche topics that might perhaps be of interest to a very small number of users. Some blogs are even open versions of personal journals which may be interesting to only a small subset of readers most likely to be close friends and family of the author. The open, unrestricted format of blogs means that the user is now able to express themselves and freely air opinions. From a business intelligence or market research perspective, this is potentially valuable data. Knowing what users think and say about your product can help better understand user preferences, likes and dislikes. Opinion retrieval is thus an important application of social media analysis. As part of this research, we have built an opinion retrieval system and participated in the TREC conference's blog track. The goal of this track was to build and evaluate a retrieval system that would find blog posts that express some opinion (either positive or negative) about a given topic or query word.

[4] http://news.bbc.co.uk
[5] http://www.cnn.com

The BlogVox system [90] that was initially built for participation at the TREC conference has spun off into a number of further applications. This framework was used to build a political analysis engine PolVox, that monitors the political blogosphere and finds opinionated posts from democrats and republicans on various topics and candidates. Further, BlogVox has resulted in the development of novel techniques for identifying trust and influence in online social media systems. Using the sentiment information around the links Kale et al. [96] use the notion of *'link polarity'* to compute the positive or negative sentiment associated with each link. This sentiment information was used to classify blogs and main stream media sources in political domain with a high accuracy. The Chapter III. of this dissertation is dedicated to social media content analysis and outlines both the semantic analysis system and the opinion retrieval system.

During the course of this research, there were a number of new trends and unexpected applications that emerged in the social media landscape. One important development was that of microblogs. Microblogging is a new form of communication in which users describe their current status in short posts distributed by instant messages, mobile phones, email or the Web. What is remarkably different about microblogging is the instantaneous nature of content and social interactions. If Wikipedia is described as our collective wisdom microblogging can be thought of as our collective consciousness. In order to sufficiently understand the nature of this trend, we crawled and analyzed a large collection of microblog updates from the site Twitter. This is the first study [94] in the literature that has analyzed the microblogging phenomenon. We find that while a number of updates tend to be of the form of daily updates, users also find such tools beneficial to share links, comment on news and seek information and quick answers from their peers.

Here, we present how to utilize the special structure of social media and the nature of social graphs to develop efficient algorithms for community detection. Several community detection approaches discussed in the literature are computationally expensive and often cubic in the number of nodes in a graph. Clearly, for the scale of social graphs and Web graphs, these algorithms are intractable. We present a novel approach to community detection using the intuition that social graphs are extremely sparse. Moreover, many properties like the degree distributions and PageRank scores follow a power-law. In such networks, a few nodes get the most attention (or links) while a large number of nodes are relatively sparsely connected. This led to the development of a novel strategy for selectively sampling a small number of columns from the original adjacency matrix to recover the community structure of the entire graph. The advantage of this approach compared to other dimensionality reduction techniques like SVD or matrix factorization methods is that it significantly reduces both the memory requirement and computation time.

One important property of social media datasets is the availability of tags. Tags or folksonomies, as they are typically called, are free-form descriptive terms that are associated with any resource. Lately, folksonomies have become an extremely popular means to organize and share information. Tags can be used for videos, photos or URLs. While structural analysis is the most widely used method for community detection, the rich meta-data available via tags can provide additional information that helps group related nodes together. However, techniques that combine tag information (or more generally content) with the structural analysis typically tend to be complicated. We present a simple, yet effective method that combines the meta-data provided by tags with structural information from the graphs to identify communities in social media. The main contribution of this technique is a simplified and intuitive approach to combining tags and graphs. Further, it achieves significant results while reducing the overhead required in processing large amount of text. Chapter IV. of this thesis outlines the structural analysis of social graphs.

Chapter V. focuses on the user perspective by analyzing feed subscriptions across a large population of users. We analyze the subscription patterns of over eighty three thousand publicly listed Bloglines[6] users. According to some estimates, *"the size of the Blogosphere continues to double every six months"* and there are over seventy million blogs (with many that are actively posting). However, our studies indicate that of all these blogs and feeds, the ones that *really matter* are relatively few. What blogs and feeds these users subscribe to and how they organize their subscriptions revealed interesting properties and characteristics of the way we consume information. For instance, most users have relatively few feeds in their subscriptions, indicating an inherent limit to the amount of attention that can be devoted to different channels. Many users organize their feeds under user-defined folder names. Aggregated across a large number of users, these folder names are good indicators of the topics (or categories) associated with each blog. We use this *collective intelligence* to measure a readerhsip-based influence of each feed for a given topic. The task of identifying the most relevant feed for a given topic or query term is now known as the *"'feed distillation task"* in the literature. We describe some applications that benefit from aggregate analysis of subscriptions including feed recommendation and influence detection.

C. Thesis Contributions

Following are the main contributions of this thesis:

- We provide a systematic study of the social media landscape by analyzing the content, structure and

[6] http://www.bloglines.com

special properties.

- Developed and evaluated innovative approaches for community detection.
  - We present a new algorithm for finding communities in social datasets.
  - SimCut, a novel algorithm for combining structural and semantic information.
- First to comprehensively analyze two important social media forms
  - We analyze the subscription patterns of a large collection of blog subscribers. The insights gained in this study were critical in developing a blog categorization system, a recommendation system as well as provide a basis for further, recent studies on feed subscription patters.
  - We analyze the microblogging phenomena and develop a taxonomy of user intentions and types of communities present in this setting.
- Finally we have built systems, infrastructure and datasets for the social media research community.

Chapter II.

BACKGROUND AND RELATED WORK

Social media research covers a broad range of topics and has fueled interest and enthusiasm from computer scientist, computational linguists to sociologists and psychologists alike. In this chapter we discuss some of the background and related work in the scope of our primary question: *"how can we analyze the structure and content of social media data to understand the nature of online communication and collaboration in social applications?"*.

A. The Social Web

The World Wide Web today has become increasingly social. In the recent book titled *"Here Comes Everybody: The Power of Organizing Without Organizations"* [187], author Clay Shirky talks about how *"personal motivation meets collaborative production"* on the Web today. One striking example is that of Wikipedia. A large number of edits in Wikipedia are minor corrections like fixing typos or adding external references. The few people who contribute the most are often driven by their passion for the subject or an altruistic motive to contribute to something useful and important. Even though each of us have different motivations behind editing a Wikipedia entry, the net effect of all these edits is a massively collaborative exercise in content production. This effort has led to creation of over 2 Million Wikipedia articles as of date and its overall size outnumbers the expensive, editorial-based encyclopedias like Encarta. This is one example of a powerful phenomena that is driving how most of the content is produced on the Web today. According to recent estimates, while editing content like CNN or Reuters news reports are about 2G per day, user generated content produced today is four to five times as much.

So, what makes the Web "*social*"? For as long as the Web has existed, content production and distribution has been one of its primary purposes. While the simplest way to create content is by editing and publishing HTML documents, blogging tools and platforms have made it much easier for an average user to *click and publish*. New tools have lowered the barrier for content production and blogs have played an important role in making it mainstream.

However, production of content alone isnt what makes the Web social. Most websites and homepages that exist are a one-way communication medium. Blogs and social media sites changed this by adding functionality to comment and interact with the content – be it blogs, music, videos or photos. The embedded social network in most applications today, along with freely edit articles and provisions to post comments is what has led to the Social Web phenomena.

Finally, the ability to connect to other users via shared resources like tags and user ratings has made it possible to find new information and like-minded individuals on the Web. Most social media sites today also have underlying recommendation systems that aid social connections and increase the findability of new information. All these factors have led to making the Web a social platform.

1. The Blogosphere

In recent years there has been an interest in studying the overall structure and properties of the Social Web. The blogosphere constitutes an important part of the Social Web. There are a number of studies that have specifically analyzed its structure and content. The blogosphere provides an interesting opportunity to study social interactions. Blogging provides a channel to express opinions, facts and thoughts. Through these pieces of information, also known as *memes*, bloggers influence each other and engage in conversations that ultimately lead to exchange of ideas and spread of information. By analyzing the graphs generated through such interactions, we can answer several questions about the structure of the blogosphere, community structure[127], spread of influence [92], opinion detection [90] and formation, friendship networks [8, 38] and information cascades [124].

In terms of size, though it constitutes only a portion of the whole Web, the blogosphere is already quite significant and is getting increasingly bigger. As of 2006 there were over 52 million blogs and presently there are in excess of 70 million blogs. The number of blogs are rapidly doubling every six months and a large fraction of these blogs are active. It is estimated that blogs enjoy a significant readership and according to the recent report by Forrester Research, one in four Americans read blogs and a large fraction of users also

participate by commenting [25]. Figure 1 shows the overall growth of the blogosphere. The current trends are only indicators of sustained growth of user-generated content.

Blogs are typically published through blog hosting sites or tools like Wordpress[1] that can be self-hosted. An entry made by a blogger appears in a reverse chronological order. Whenever a new post is published, a ping server is notified of the fresh content. Infrastructurally, this is one of the critical difference from the Web. While on the Web, search engines rely on crawlers to fetch and update the index with new content, the stream of pings provides information that new content has been published on a blog. This is done essentially to ensure that downstream services (like search engines and meme trackers) can quickly find new content, thus ensuring the freshness of their index.

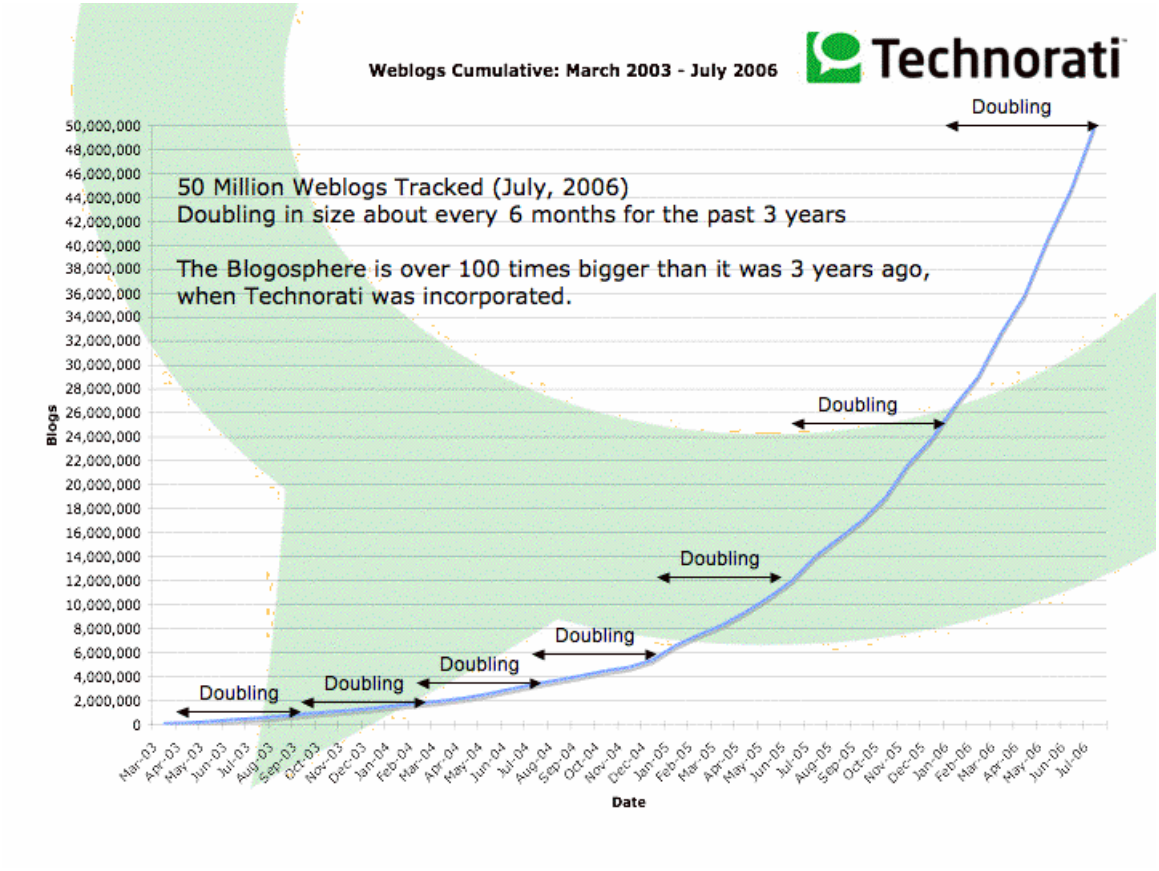
The blog home page can contain various anchortext links that provide personal information, links to recent posts, photos, blogrolls (links to blogs frequently read), delicious bookmarks, FOAF descriptions etc. Each blog post contains a title, date, time and the content of the post. Additionally, posts can also be assigned tags or categories that provide information about the topic or keywords that are relevant to the post. Finally the blog itself can be subscribed via RSS (Really Simple Syndication) feeds. Through this simple XML formatted file, users can subscribe to blogs, news sites and also personalized content such as alerts and search results.

2. Social Networking Sites

In the book *"Click: What Millions of People are Doing Online and Why It Matters"* [195], author Bill Tancer discusses how social networking sites today attract the highest traffic on the internet today. With hundreds of social networking sites specializing in different niches, users can connect with people sharing similar interests and also keep in touch with ex-colleagues, classmates, friends and family. Social networking sites cater to a wide variety of audience from teens (MySpace) to college students (Facebook) to professional networks (LinkedIn).

One implication of the widespread usage of these sites is privacy concerns. Several researchers have focused on studying the usage patterns and performed longitudinal studies of users on these networks. This has been of interest to both computer scientists and social scientists alike. In a recent study of Facebook users, Dr. Zeynep Tufecki concluded that Facebook users are very open about their personal information [198, 199]. A surprisingly large fraction openly disclose their real names, phone numbers and other personal information.

[1] http://www.wordpress.org



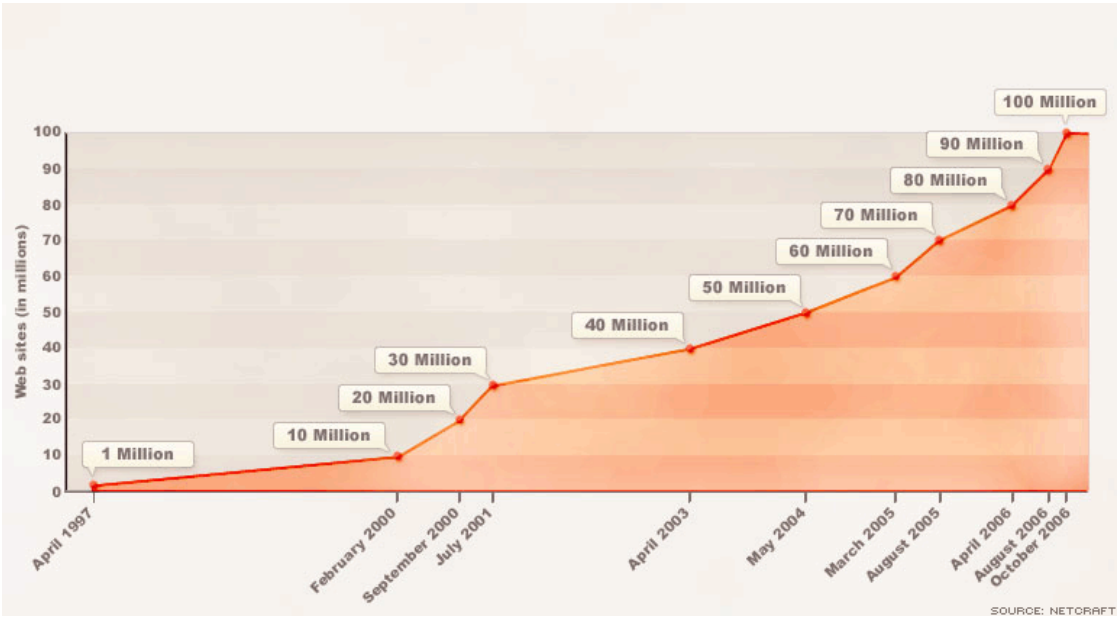


Figure 1: *The blogosphere continues to double every six months. This increase has also contributed to the growth of the Web in general (sources: Technorati, Netcraft, CNN)*

In his research as well as the recent book *Snoop: What your stuff says about you?* [60], Dr. Sam Gosling talks about how personal spaces like bedrooms, office desks and even Facebook profiles reveal a whole lot about the real self. Their research [178] indicates how using just the information from a Facebook profile page, users can accurately score openness, conscientiousness, extraversion, agreeableness, and neuroticism (also known as the five factor model in Psychology).

3. Tagging and Folksonomies

The term folksonomy refers to free-form tags or descriptive words associated with a resource like a URL, document or a video. This form of meta-data information has been popularized by social bookmarking sites like delicious[2] and photo sharing sites like flickr[3] and it provides a convenient way for users to organize information and share it with their social network. The term *folksonomy* is derived from folk and taxonomy and is attributed to Thomas Vander Wal.

In a recent paper, Heymann et al. [76] inquire the effectiveness of tagging and applications of social bookmarking in Web search. This extensive study of del.icio.us finds that the social bookmarking and tagging is a growing phenomena. While the index of URLs available in a bookmarking site like del.icio.us is much smaller compared to the overall size of the Web, this study indicates that important sites are well represented in such systems. Brooks and Montanez [21] have also studied the phenomenon of user-generated tags and evaluate effectiveness of tagging. In contrast, Chi et al. [26] find that as more users join the bookmarking site and annotate more documents with free-form tags, the efficiency of tagging is in fact decreasing.

Tagging is essentially a means to organize information and provide an easy way to organize and share information collaboratively. Despite large differences in motivations of tagging and usage of tags, a stable consensus emerges [59]. Studies have also shown that simple stochastic models of tagging can explain user behavior in such environments. Cattuto et al. [22] model users as simple agents that tag documents with a frequency-bias and have the notion of memory, such that they are less likely to use older tags. Surprisingly, this simple explanation of user behavior is quite accurate in modeling how we use tags.

Dubinko et al. [46] describe tag visualization techniques by using Flickr tags. Their work concentrates on automatically discovering tags that are most 'interesting' for a particular time period. By visualizing these on a timeline they provide a tool for exploring the usage and evolution of tags on Flickr. Several techniques for 'tag recommendations' have been proposed in recent years. AutoTagging [146] is a collaborative filtering

[2] http://del.icio.us
[3] http://www.flickr.com

based recommendation system for suggesting appropriate tags. Heymann et al. [77] and Zhou et al. [210] present techniques for predicting and recommending tags. TagAssist [192], is a system that recommends tags related to a given blog post.

All these systems demonstrate several applications of tagging and folksonomies. In context to this research, we present an analysis of tag usage through folder names. We analyze a large collection of users and the organization of their feed subscriptions. Categorizing feeds under folder names is a common practice among users and it gives us a way to group related feeds. We describe applications of our analysis in feed distillation and recommendation. The second way in which we incorporate tag information is by studying the use of tagging in clustering graphs. We demonstrate that tags can provide additional information that is useful in grouping related blogs and can improve clustering results over graph-only methods.

B. Mining the Social Web

1. Structure and Properties

A number of researchers have studied the graph structure of the Web. According to the classic 'Bow Tie' model [18] the WWW exhibits a small world phenomenon with a relatively large portion of links constituting the core or Strongly Connected Component (SCC) of the graph. Ravi Kumar et. al. [116] have studied the evolution of the blog graph and find that the size of the blogosphere grew drastically in 2001. They find that at a microscopic level there was also emergence of stronger community structure. There have been further research that has analyzed the structure of the blogosphere and compared its statistical parameters to those of the Web.

Currently, there are two large samples of the blogosphere that are available for researchers. One of them is a collection used for the WWE 2006 workshop that consists of a collection of blogs during a three week period during the year 2005. The second collection is the TREC 2006 dataset [131], which is over a 11 week period that consists of blogs that were crawled starting from a small subset. A recent paper by Shi et al. [184] surveys these datasets and compares them to the known parameters of the Web. Interestingly, inspite of the the sparsity of data, there are a lot of similarities of the blog graphs with the Web graphs. Both datasets show power-law slopes of around 2 to 2.5 which is very close to the 2.1 observed in the Web. Similar values are also corroborated by Kumar et al. [116] in their study. Using a graph represented by the link structure of the blog post to blog post links from a collection of about 3 Million blogs we find power law distributions

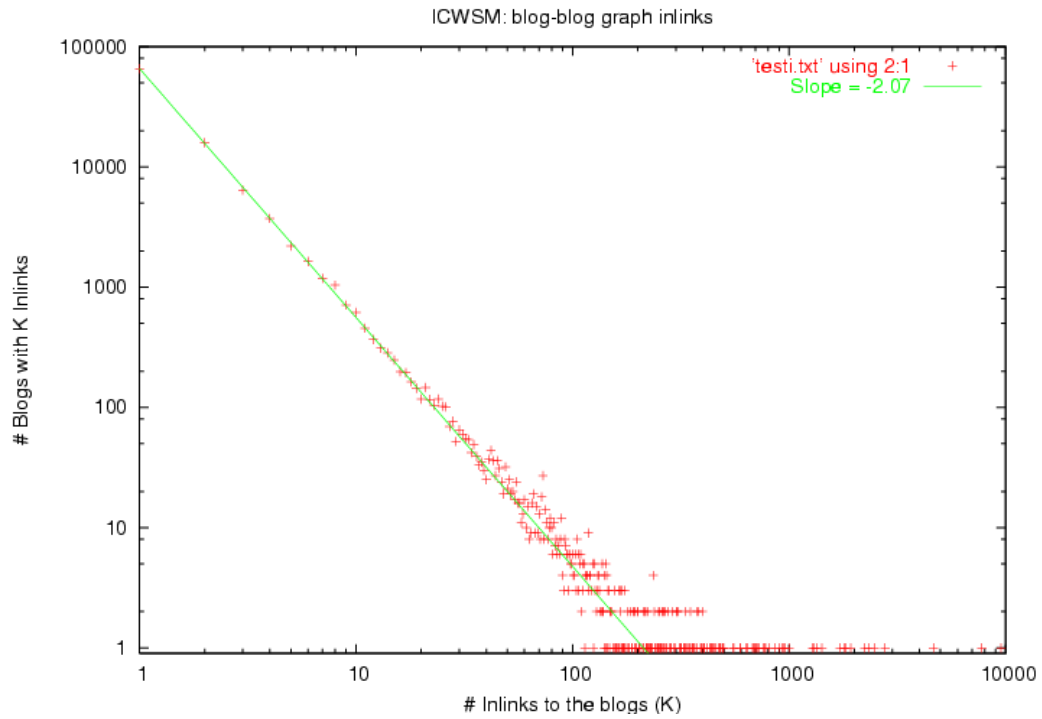


Figure 2: Inlink Distribution for ICWSM dataset

for both the inlink and outlinks in the graph (see Figure 2). Similar results were also discussed in [68] while studying how topics propagate on the blogosphere.

Leskovec et al. [123] present the "Forest Fire" model to explain the growth and evolution of dynamic social network graphs. There are two theories that support this model. First is the "desnification of power law" that states that the out degree increases over time as the networks evolve and the second is the "shrinking diameter" according to which the average diameter of a network decreases over time. As this is a generative process the goal is to build simulated graphs that have properties that closely match those of the real world. The forest fire model tries to mimic the way information spreads in networks. In this model, new nodes arrive one at a time and attach themselves to an existing node preferentially. Once the node is connected, it performs a random walk in the neighborhood and creates new links locally. The process is them repeated for each of the new nodes that are linked to during the random walk. The forest fire model was also shown to describe information cascades in blog graphs [124]. Information cascades are a chain of links from one blog to another that describe a conversation. Interestingly, the authors find that the distribution of the sizes of such cascades also follow a power law distribution.

In a related work, Karandikar and Java et al. [98] present a generative model that accurately models several properties of the blog graphs, including the distributions of the blog to blog network as well as the statistics of the post to post network. This model builds on the existing approaches by presenting a simple behavioral model in which a blogger is treated as both a reader and a writer. When a blogger is in a read mode, she performs a random walk in the neighborhood of the blog and links to recently read posts, when transitioning into the write mode. The advantage of this model is that it generates the synthetic blog to blog network by modeling the behavior of a user that results in creation of new posts in the graph.

In Chapter V., we present an analysis of readership patterns of a large collection of users. These patterns allow us to gain an understanding into the reading habits of a number of users and also provides an intuitive way to organize feeds into a topic hierarchy.

2. Mining Social Media Content

Content on blogs may be quite different from that of the Web. Blogs tend to be more personal, topical and are often emotionally charged. Blogs and online journals are a reflection of our thoughts, opinions and even moods [143]. The TREC conference's blog track has focussed opinion retrieval [164]. This task involves, identifying blog posts that express positive or negative sentiment about a given query term has been a topic of significant interest. In Chapter III., we present the BlogVox system that was built for this task.

Analyzing blog content can also have a number of potential business intelligence and advertising applications. Kale et al. [96] present an approach to use sentiment information for classifying political blogs. Mishne et al. [145] describe how mentions of movie names combined with the sentiment information can be correlated with its sales. Such applications have a financial incentive and provide important insights into markets and trends. Content analysis also proves useful in advertising and marketing. Mishne et al. [148] also present a technique for deriving "wishlists" from blogs and identify books that might be of potential interest to a blogger, based on the content of the posts. Finally, language models built using the blog posts and special features like tags is also shown to have effective results in matching relevant ads[144].

Herring et al. [74] performed an empirical study the interconnectivity of a sample of blogs and found conversations on the blogosphere are sporadic and highlight the importance of the 'A-list' bloggers and their roles in conversations. A-list bloggers are those that enjoy a high degree of influence in the blogosphere. These are the blogs that correspond to the head of the long tail (or power-law) distribution of the blogosphere. As shown in figure 2., these consitute a small fraction of all the blogs that receive the most attention or

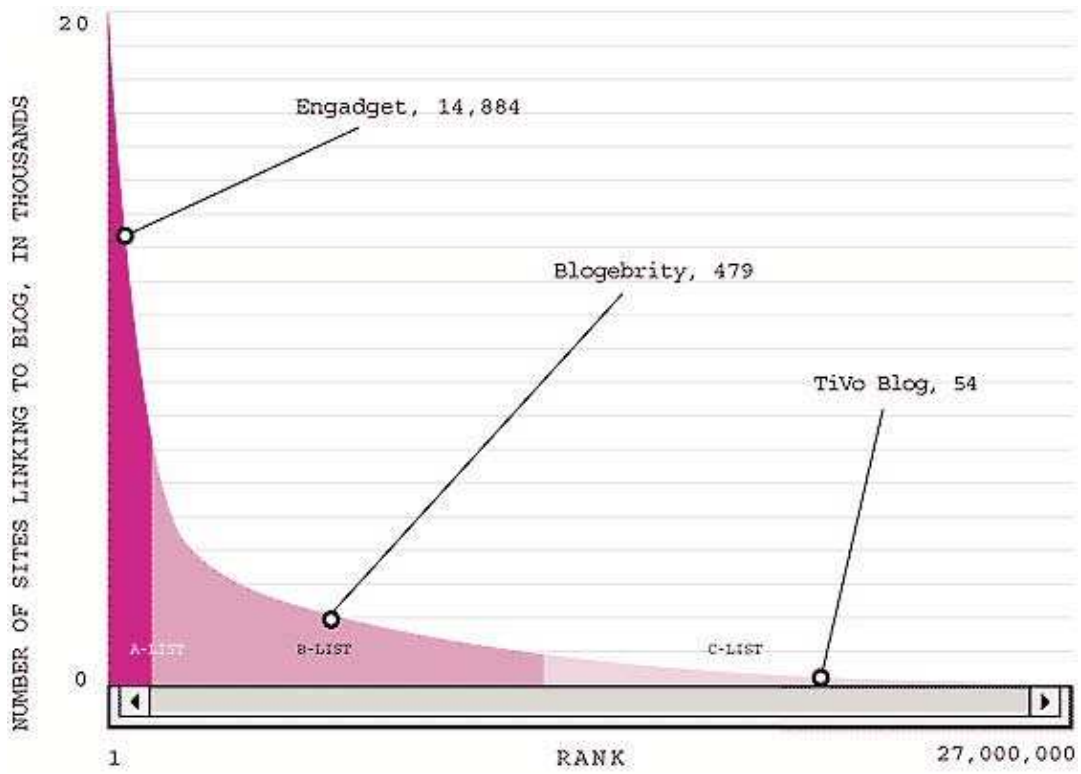


Figure 3: The long tail distribution. Typically a nodes will receive the most attention. This is also popularly known as the 80-20 distribution. (Source NYT)

links. This type of distribution has become synonymous many different social datasets. Blog search engine Technorati lists the top 100 blogs on the blogosphere. These lists, while serving as a generic ranking purpose, do not indicate the most popular blogs in different categories. This task was explored by Java et al. [88] to identify the "*Feeds that Matter*". The TREC 2007 blog track [132] defines a new task called the feed distillation task. Feed distillation, as defined in TREC 2007 is the task of identifying blogs with recurrent interest in a given topic. This is helpful for example, in allowing the user to explore interesting blogs to subscribe for a given topic. Elsas et al. [48] explored two approaches to feed distillation. One approach is to consider the entire feed as a single document. The retrieval task was to find the most relevant documents (i.e feeds) in this setting. They used Wikipedia as a resource for query expansion to help identify relevant terms for a given topic. The second model is to identify the posts that are relevant and find feeds that correspond to the most relevant posts returned. They find that the modeling the feed itself as a document is more appropriate for this task.

A related task is that of identifying influential nodes in a network. There are several different interpreta-

tions of what makes a blog or a node in a social network influential. Song et al. [191] predict and rank nodes in a network based on the flow of information. Their proposed algorithm, DiffusionRank identifies the most likely individuals in a network who most likely to receive a given information. Some models for maximizing such a flow of information is proposed by Kempe et al. [102, 103]. They use a greedy heuristic based approach for identifying the set of nodes that are capable of influencing the largest fraction of the network. InfluenceRank [190] is an algorithm similar to PageRank that is used to identify the opinion leaders in the blogosphere. This approach is based on content analysis of the blog post and the outlinks that they point to. The intuition is that those providing novel information are more likely to be opinion leaders in such networks. In Chapter V., we present a novel approach to detect influential nodes. We use a combination of link analysis and feed readership information for identifying the most influential blogs in different topics.

3. Communities in Social Graph

Social structure in any society emerges from our desire to connect with others around us who share similar views and interest. Communities emerge in many types of networks. Starting with Milgram's experiments [141] that led to the popular anecdote on the *'six degrees of separation'*, the study of the underlying structure and properties has interested researchers for many years. Many real world networks like collaboration/coauthor [154], biological networks [203] and internet exhibit the small-world phenomenon.

Flake et. al. [2] describe a network flow based approach to partitioning the graph into communities. Recently, there has been renewed interest in community detection for blog data. Lin et. al. [127] identify a group of blogs that are mutually aware of each other. Post-to-post links, comments, trackbacks, all constitute to different types of actions that indicate awareness. Using an algorithm similar to PageRank each pair of blogs is weighted with an association score based on the different actions between the corresponding blogs. However, this technique requires a seed set of blogs to extract the community. Additionally, they provide a clustering algorithm to visualize such communities [197].

Some community detection techniques require computation of "betweenness centrality" which is an expensive calculation over very large graphs [160]. Betweenness centrality is a measure of the number of times a node is on the shortest path route amongst all other pairs of nodes. Newman provides a fast approximation [155] to this measure. Figure 4 shows a visualization of an example community of political blog graph, identified using this approach. The size of the node is proportional to the degree of the blog.

While several researchers have studied static networks, most real-world networks are temporal and dy-

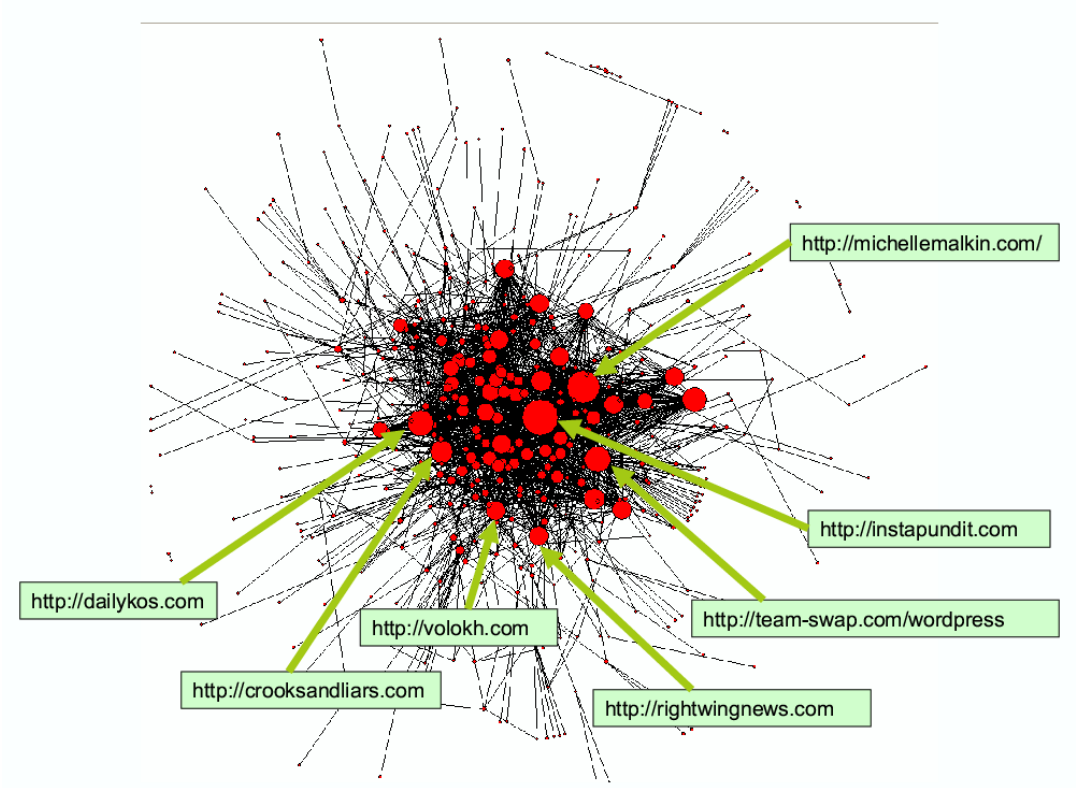


Figure 4: *A view of a sub-community containing a number of political blogs consisting about 13K vertices. The size of the node is proportionate to its degree.*

namic in nature. Communities form through interactions over a long period of time and they change due to shifts in interests, community member's reactions to News events and factors. Communities may merge to form a larger community or a single community may split into a number of smaller groups. Chi et al. [27] extend the spectral clustering algorithms for evolving social network graphs and blog graphs. Chi et al. [28] also present a different approach to community detection that is based on both the structural and temporal analysis of the interactions between nodes. A community is understood to be a set of nodes that interact more closely with each other and this is captured by the structural analysis. However, there is a second component to communities which is the sustained interaction or interest between nodes over time. This is accounted for by considering the temporal nature of these interactions. Their method is based on factorizing the tensor matrix that captures interactions between nodes over time. A further extension of this technique is presented by Lin et al. [126].

In context of this work, we present two techniques for community analysis. Most of the existing approaches to community detection are based on link analysis and ignore the folksonomy meta-data that is easily available on in social media. We present a novel method to combine the link analysis for community detection with information available in tags and folksonomies, yielding more accurate communities. Many social graphs can be quite huge. In the second part of our community detection work we focus on effectively sampling a small portion of the graph in order to approximately determine the overall community structure. These techniques are discussed in Chapter IV. of this dissertation.

Chapter III.

MINING SOCIAL MEDIA CONTENT

Social media content, especially blogs, often consists of noisy, ungrammatical and poorly structured text. This makes open domain tasks like opinion retrieval and classification for blogs quite challenging. In addition any text analytics system that deals with blogs must address two key issues: (i) detecting and eliminating spam blogs and spam comments and (ii) eliminating noise from link-rolls and blog-rolls. In this Chapter we discuss the BlogVox opinion retrieval system. We describe a framework that indexes a large collection of blogs and provides an interface for finding opinionated blog posts that express some sentiment (either positive or negative) with respect to given query terms. In such an application some of the data cleaning issues mentioned above play a critical role in ensuring high quality results. We also discuss the various scoring mechanisms for sentiment ranking.

The second part of this chapter concerns deeper semantic processing of social media content. While the BlogVox opinion retrieval system was mostly syntactic and uses shallow parsing and lexicon-based approaches, SemNews is a semantic news framework that is capable of large scale semantic processing. The infrastructure has the capability of indexing several thousands of news feeds and processing the summaries of news articles to extract the meaning representation of the stories. This provides a capability to process and make text machine readable. SemNews uses a sophisticated natrual language processing engine that is supported with an extensive ontology. The extracted meaning representation of the stories are exported in Semantic Web language OWL.

A. Mining Sentiments and Opinions

The BlogVox system retrieves opinionated blog posts specified by ad hoc queries. BlogVox was developed for the 2006 TREC blog track by the University of Maryland, Baltimore County and the Johns Hopkins University Applied Physics Laboratory using a novel system to recognize legitimate posts and discriminate against spam blogs. It also processes posts to eliminate extraneous non-content, including blog-rolls, link-rolls, advertisements and sidebars. After retrieving posts relevant to a topic query, the system processes them to produce a set of independent features estimating the likelihood that a post expresses an opinion about the topic. These are combined using an SVM-based system and integrated with the relevancy score to rank the results. We evaluate BlogVox's performance against human assessors. We also evaluate the individual splog filtering and non-content removal components of BlogVox.

The BlogVox system was developed by the University of Maryland, Baltimore County and the Johns Hopkins University Applied Physics Laboratory to perform the opinion retrieval task defined by the 2006 TREC Blog Track. In this task, a user enters a query for a topic of interest (e.g., March of the Penguins) and expects to see a list of blog posts that express an opinion (positive or negative) about the topic. The results are ranked by the likelihood that they are expressing an opinion about the given topic. The approach used in BlogVox has several interesting features. Two techniques are used to eliminate spurious text that might mislead the judgment of both relevance and opinionatedness. First, we identify posts from spam blogs using a machine-learning based approach and eliminate them from the collection. Second, posts are "cleaned" before being indexed to eliminate extraneous text associated with navigation links, blog-rolls, link-rolls, advertisements and sidebars. After retrieving posts relevant to a topic query, the system applies a set of scoring modules to each producing a vector of features estimating the likelihood that a post expresses an opinion about the topic. These are combined using an SVM-based system and integrated with the overall relevancy score to rank the results.

Opinion extraction and sentiment detection have been previously studied for mining sentiments and reviews in domains such as consumer products [37] or movies [167, 52]. More recently, blogs have become a new medium through which users express sentiments. Opinion extraction has thus become important for understanding consumer biases and is being used as a new tool for market intelligence [57, 161, 129].

Blog posts contain noisy, ungrammatical and poorly structured text. This makes open-domain, opinion retrieval for blogs challenging. In addition any text analytics system that deals with blogs must address two larger issues: (i) detecting and eliminating posts from spam blogs (commonly known as splogs) and spam

comments and (ii) eliminating irrelevant text and links that are not part of the post's content.

Recently, Spam blogs, or splogs have received significant attention, and techniques are being developed to detect them. Kolari, et al. [108] have recently discussed the use of machine learning techniques to identify blog pages (as opposed to other online resources) and to categorize them as authentic blogs or spam blogs (splogs). [111] extends this study by analyzing a special collection of blog posts released for the Third Annual Workshop on the Weblogging Ecosystem held at the 2006 World Wide Web Conference. Their findings on spam blogs confirms the seriousness of the problem, the most recent data shows about 64% of "pings" collected from the most popular ping-server for English blogs are from splogs.

Blog posts are complex documents and consist of a core containing the post's real content surrounded by an array of extraneous and irrelevant text, images and links. This "noise" includes links to recent posts, navigational links, advertisements and other Web 2.0 features such as tag rolls, blog rolls, Technorati tags, Flickr links and often accounts for 75% or more of the post's size. The presence of this extra material can make it difficult for text mining tools to narrow down and focus on the actual content of a blog post. Moreover, these features may also reduce search index quality. Discounting for such noise is especially important when indexing blog content. Blog posts are complex documents and consist of a core containing the post's real content surrounded by an array of extraneous and irrelevant text, images and links. This "noise" includes links to recent posts, navigational links, advertisements and other Web 2.0 features such as tag rolls, blog rolls, Technorati tags, Flickr links and often accounts for 75% or more of the post's size. The presence of this extra material can make it difficult for text mining tools to narrow down and focus on the actual content of a blog post. Moreover, these features may also reduce the quality of the search index. Discounting for such noise is especially important when indexing blog content.

1. Related Work

Different sentiment classification techniques have been applied in movies and product domains. Many of these techniques use a combination of machine learning, NLP and heuristic techniques. While some of the work looks at identifying opinions at a document level, others have tried to classify sentences and summarize opinions.

Most effective among the machine learning algorithms are naive bayes, SVM. These are mainly used to learn recognize either linguistic patterns that are indicators of opinions or sentiment bearing words and phrases. Turney [200] proposed the application of unsupervised machine learning algorithm for sentiment

classification by comparing the orientation of the phrase with the terms ‘excellent’ and ‘poor’.

Minqing Hu and Bing Liu [82] propose using WordNet to determine the polarity of different adjectives. Their goal is to identify sentiment at a sentence level. The the overall polarity score for a sentence is determined by combining the weights contributed by each of the adjectives near a feature word. The Opinion Observer system [129] extends this work to summarizing the pros and cons of various features of a product.

Tracking sentiment change over time has been studied by Tong [196] and more recently in the context of blogs [52].

2. The TREC Blog Track

The 2006 TREC Blog track, organized by NIST, asked participants to implement and evaluate a system to do “opinion retrieval” from blog posts. Specifically, the task was defined as follows: build a system that will take a query string describing a topic, e.g., “March of the Penguins”, and return a ranked list of blog posts that express an opinion, positive or negative, about the topic.

For training and evaluation, NIST provided a dataset of over three million blogs drawn from about 80 thousand blogs. The TREC dataset consisted of a set of XML formatted files, each containing blog posts crawled on a given date. The entire collection consisted of over 3.2M posts from 100K feeds [131]. These posts were parsed and stored separately for convenient indexing, using the HTML parser tool [1]. Non-English blogs were ignored in addition to any page that failed to parse due to encoding issues.

In order to make the challenge realistic NIST explicitly included 17,969 feeds from splogs, contributing to 15.8% of the documents. There were 83,307 distinct homepage URLs present in the collection, of which 81,014 could be processed. The collection contained a total of 3,214,727 permalinks from all these blogs.

TREC 2006 Blog Track participants built and trained their systems to work on this dataset. Entries were judged upon an automatic evaluation done by downloading and running, without further modification to their systems, a set of fifty test queries.

3. BlogVox Opinion Retrieval System

Compared to domain-specific opinion extraction, identifying opinionated documents about a randomly chosen topic from a pool of documents that are potentially unrelated to the topic is a much more difficult task. Our goal for this project was to create a system that could dynamically learn topic sensitive sentiment words

[1] http://htmlparser.sourceforge.net/

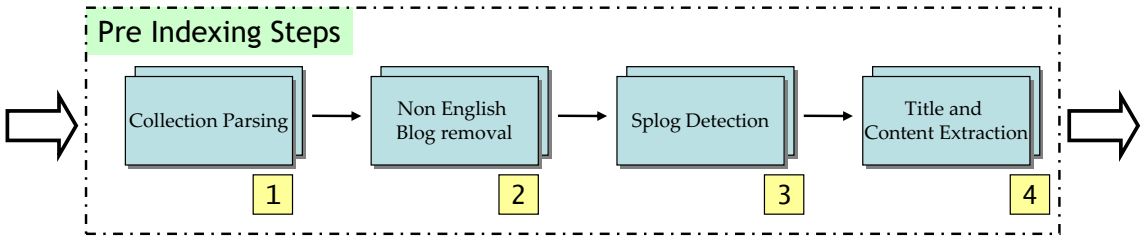


Figure 5: BlogVox text Preparation steps: 1. Parse the TREC corpus 2. Remove non English posts 3. Eliminate splogs from the collection 4. Remove spurious material from the DOM tree.

to better find blog posts expressing an opinion about a specified topic. After cleaning the TREC 2006 Blog Track dataset in the pre-indexing stage, blog posts are indexed using Lucene, an open-source search engine. Given a TREC query BlogVox retrieves a set of relevant posts from the Lucene index and sends the posts to the scorers. Using a SVM BlogVox ranks each document based upon the score vector generated for the document by the set of scorers show in Figure 6.

We tuned Lucene's scoring formula to perform document length normalization and term specific boosting [2]. Lucene internally constructs an inverted index of the documents by representing each document as a vector of terms. Given a query term, Lucene uses standard Term Frequency (TF) and Inverse Document Frequency (IDF) normalization to compute similarity. We used the default parameters while searching the index. However, in order to handle phrasal queries such as "United States of America" we reformulate the original query to boost the value of exact matches or proximity-based matches for the phrase.

4. Data Cleaning

Two kinds of spam are common in the blogosphere (i) spam blogs or splogs, and (ii) spam comments. We first discuss spam blogs, approaches on detecting them, and how they were employed for BlogVox.

Identifying and Removing Spam

Splogs are blogs created for the sole purpose of hosting ads, promoting affiliate sites (including themselves) and getting new pages indexed. Content in splogs is often auto-generated and/or plagiarized, such software sells for less than 100 dollars and now inundates the blogosphere both at ping servers (around 75% [107]) that monitor blog updates, and at blog search engines (around 20%, [112]) that index them. Spam comments pose an equally serious problem, where authentic blog posts feature auto-generated comments that target

[2] http://lucene.apache.org/java/docs/scoring.html

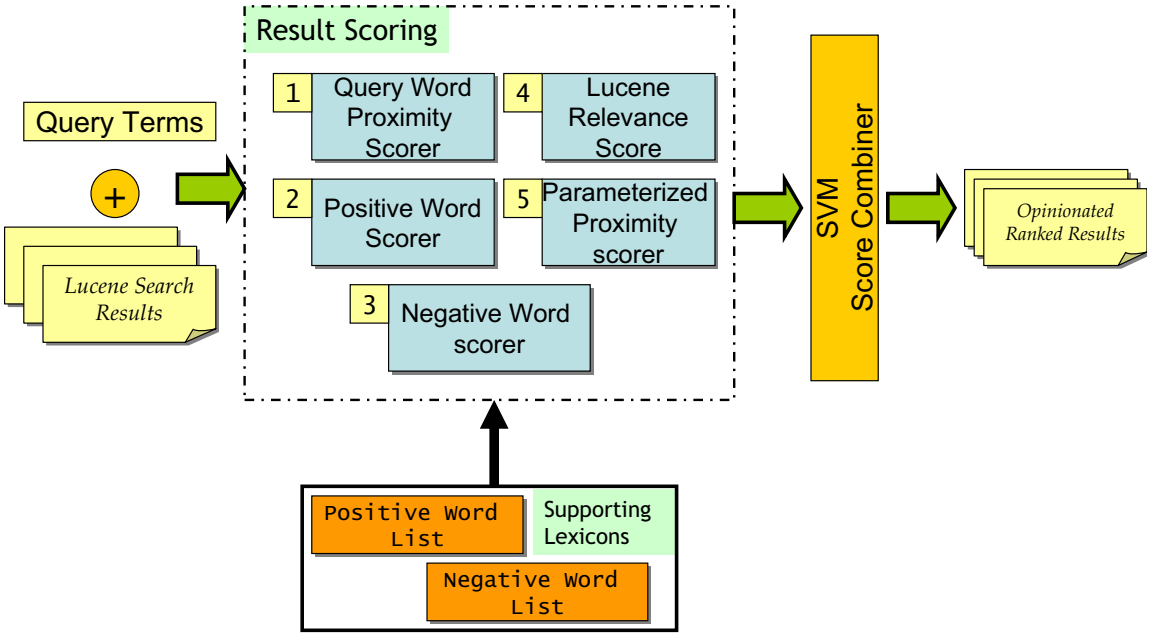


Figure 6: After relevant posts are retrieved, they are scored by various heuristics and an overall measure of oinionatedness computed by a SVM.

Figure 7: A typical splog, plagiarizes content (ii), promotes other spam pages (iii), and (i) hosts high paying contextual advertisements

ranking algorithms of popular search engines. A popular spam comment filter [3] estimates the amount of spam detected to be around 93%.

Figure 7 shows a splog post indexed by a popular blog search engine. As depicted, it features content plagiarized from other blogs (ii), displays ads in high paying contexts (i), and hosts hyperlinks (iii) that create link farms. Scores of such pages now pollute the blogosphere, with new ones springing up every moment. Splogs continue to be a problem for web search engines, however they present a new set of challenges for blog analytics. Splogs are well understood to be a specific instance of the more general spam web-pages [71]. Though offline graph based mechanisms like TrustRank [72] are sufficiently effective for the Web, the blogosphere demands new techniques. The quality of blog analytics engines is judged not just by content coverage, but also by their ability to quickly index and analyze recent (non-spam) posts. This requires that fast online splog detection/filtering [108][177] be used prior to indexing new content.

We employ statistical models to detecting splogs as described by [112], based on supervised machine learning techniques, using content local to a page, enabling fast splog detection. These models are based solely on blog home-pages, and are based on a training set of 700 blogs and 700 splogs. Statistical models based on local blog features perform well on spam blog detection. See Table III..1. The bag-of-words based features slightly outperforms bag-of-outgoingurls (URL's tokenized on '/') and bag-of-outgoinganchors. Additional results using link based features are slightly lower that local features, but effective nonetheless.

[3] http://akismet.com

Feature	*Precision*	*Recall*	*F1*
words	.887	.864	.875
urls	.804	.827	.815
anchors	.854	.807	.830

Table III..1: SVM Models with 19000 word features and 10000 each of URL and anchor text features (ranked using Mutual Information) can be quite effective for splog detection.

Interested readers are referred to [112] for further details. Therefore, BlogVox used only local features to detect splogs.

Comment spam occurs when a user posts spam inside a blog comment. Comment spam is typically managed by individual bloggers, through moderating comments and/or using comment spam detection tools (e.g. Akismet) on blogging platforms. Comment spam and splogs share a common purpose. They enable indexing new web pages, and promoting their page rank, with each such page selling online merchandise or hosting context specific advertisements. Detecting and eliminating comment spam [147] depends largely on the quality of identifying comments on a blog post, part of which is addressed in the next section.

Identifying Post Content

Most extraneous features in blog post are links. We describe two techniques to automatically classify the links into content-links and extra-links. Content links are part of either the title or the text of the post. Extra links are not directly related to the post, but provide additional information such as: navigational links, recent entries, advertisements, and blog rolls. Differentiating the blog content from its chaff is further complicated by blog hosting services using different templates and formats. Additionally, users host their own blogs and sometimes customize existing templates to suit their needs.

Web page cleaning techniques work by detecting common structural elements from the HTML Document Object Model (DOM) [207, 208]. By mining for both frequently repeated presentational components and content in web pages, a site style tree is constructed. This tree structure can be used for data cleaning and improved feature weighting. Finding repeated structural components requires sampling many web pages from a domain. Although blogs from the same domain can share similar structural components, they can differ due to blogger customization. Our proposed technique does not require sampling and works independently on each blog permalink.

Instead of mining, we used a simple general heuristic. Intuitively extraneous links tend to be tightly grouped containing relatively small amounts of text. Note that a typical blog post has a complex DOM tree

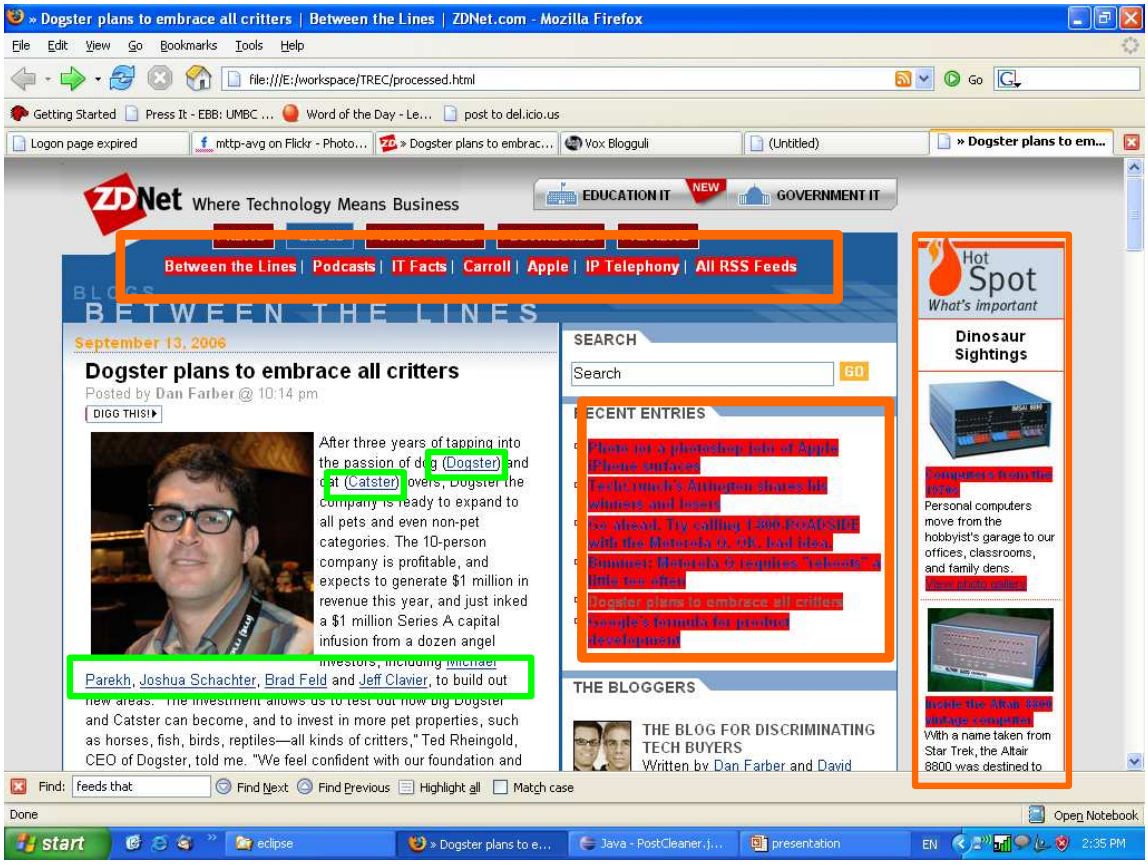


Figure 8: A typical blog post containing navigational links, recent posts, advertisements, and post content with additional links in it. Highlighted links are eliminated by the approximation heuristic.

with many parts, only one of which is the content of interest in most applications.

After creating the DOM tree we traverse it attempting to eliminate any extraneous links and their corresponding anchor text, based upon the preceding and following tags. A link **a** is eliminated if another link **b** within a θ_{dist} tag distance exists such that:

- No title tags (H1, H2...) exist in a θ_{dist} tag window of **a**.
- Average length of the text bearing nodes between **a** and **b** is less than some threshold.
- **b** is the nearest link node to **a**.

The average text ratio between the links, $\alpha_{avgText}$ was heuristically set to 120 characters and a window size, θ_{dist} of 10 tags was chosen. The Algorithm 1 provides a detailed description of this heuristic.

Next we present a machine learning approach to the link classification problem. From a large collection of blog posts, a random sample of 125 posts was selected. A human evaluator judged a subset of links

Algorithm 1 Blog post cleaning heuristic

```
Nodes[] tags = tags in the order of the depth first traversal of the DOM tree
for all i such that 0 ≤ i ≤ |tags| do
    dist = nearestLinkTag(tags, i);
    if dist ≤ θ_dist then
        eliminate tags[i]
    end if
end for
```

Procedure 2 int nearestLinkTag(Nodes[] tags, int pos)

```
minDist = |tags|
textNodes = 0
textLength = 0
title = false;
for all j such that pos − θ_dist ≤ j ≤ pos + θ_dist do
    node = tags[j]
    if j = 0||j = pos||j > (|tags| − 1) then
        continue
    end if
    if node instanceOf TextNode then
        textNodes++;
        textLength += node.getTextLength();
    end if
    dist = |pos − j|
    if node instanceOf LinkNode && dist < minDist then
        minDist = dist
    end if
    if node instanceOf TitleNode then
        title = true
    end if
end for
ratio = textLength / textCount
if ratio > α_avgText||title == true then
    return tags.size()
end if
return minDist
```

ID	Features
1	Previous Node
2	Next Node
3	Parent Node
4	Previous N Tags
5	Next N Tags
6	Sibling Nodes
7	Child Nodes
8	Depth in DOM Tree
9	Char offset from page start
10	links outside the blog?
11	Anchor text words
12	Previous N words
13	Next N words

Table III..2: Features used for training an SVM for classifying links as content links and extra links.

Method	*Precision*	*Recall*	*F1*
baseline heuristic	0.83	0.87	0.849
svm cleaner (tag features)	0.79	0.78	0.784
svm cleaner (all features)	0.86	0.94	0.898

Table III..3: Data cleaning with DOM features on a training set of 400 HTML Links.

(approximately 400) from these posts. The links were manually tagged either content-links or extra-links. Each link was associated with a set of features. Table III..2 summarizes the main features used. Using this feature set an SVM model was trained [4] to recognize links to eliminate. The first set of features (1-7) was based on the tag information. The next set of features (8-9) was based on position information and the final set of features (10-13) consisted of word-based features. Using features (1-7) yields a precision of 79.4% and recall of 78.39%, using all our features (1-13) yields a precision of 86.25% and recall of 94.31% under 10-fold cross validation.

We compared the original baseline heuristic against human evaluators. The average accuracy for the baseline heuristic is about 83% with a recall of 87%.

5. Evaluations

To improve the quality of opinion extraction results, it is important to identify the title and content of the blog post because the scoring functions and the Lucene indexing engine can not differentiate between text present in the links and sidebars from text present in content of the blog post. Thus, a post which has a link to a recent post titled ‘Why I love my iPod’ would be retrieved as an opinionated post even if the post content is about

[4] `http://svmlight.joachims.org/`

some other topic. This observation lead to the development of our first scorers.

As shown in figure 6, a number of heuristics are employed to score the results based on the likelihood that it contains an opinion about the query terms. These scorers work by using both document level and individual sentence level features. Some of the scoring heuristics were supported by a hand-crafted list of 915 generic postive and 2712 negative sentiment words.

The following is a brief description of each scoring function:

Query Word Proximity Scorer finds the average number of sentiment terms occurring in the *vicinity* of the query terms using a window size of 15 words before and after the query terms. If the query is a phrasal query, the presence of sentiment terms around the query was weighted twice.

Parametrized Proximity Scorer was similar to the Query Word Proximity Scorer. However, we used a much smaller dictionary which was divided into two subsets: highly polar sentiment words, and the relatively less polar words. We used parameters to specify the window of text to search for sentiment words (five and fifteen), and to boost sentiment terms around phrase queries (one and three). This resulted in a total of eight scorers.

Positive and Negative Scorers counted the number of sentiment words (positive, negative) in the entire post.

Lucene Relevance Score was used to find how closely the post matches the query terms.

We also experimented with other scoring functions, such as adjective word count scorer. This scorer used an NLP tool to extract the adjectives around the query terms. However, this tool did not perform well mainly due to the noisy and ungrammatical bsentences present in blogs.

Once the results were scored by these scoring modules, we used a meta-learning approach to combine the scores using SVMs. Our SVMs were trained using a set of 670 samples of which 238 were positive (showed a sentiment) and the rest were negative. Using the polynomial kernel with degree gave the best results with precision of 80% and recall of 30%. The model was trained to predict the probability of a document expressing opinion. This value was then combined with the Lucene relevance score to produce final runs.

The opinion extraction system provides a testbed application for which we evaluate different data cleaning methods. There are three criteria for evaluation: i) improvements in opinion extraction task with and without data cleaning ii) performance evaluation for splog detection iii) performance of the post content identification.

Splog Detection Evaluation

Our automated splog detection technique identified 13,542 blogs as splogs. This accounts for about 16% of the identified homepages. The total number of splog permalinks is 543,086 or around 16% of the collection, which is very close to the 15.8% explicitly included by NIST. While the actual list of splogs are not available for comparison, the current estimate seem to be close. To prevent the possibility of splogs skewing our results permalinks associated with splogs were not indexed.

Given a search query, we would like to estimate the impact splogs have on search result precision. Figure 9 shows the distribution of splogs across the 50 TREC queries. The quantity of splogs present varies across the queries since splogs are query dependent. For example, the topmost spammed query terms were ‘cholesterol’ and ‘hybrid cars’. Such queries attract a target market, which advertiser can exploit.

The description of the TREC data [131] provides an analysis of the posts from splogs that were added to the collection. Top informative terms include ‘insurance’, ‘weight’, ‘credit’ and such. Figure 10 shows the distribution of splogs identified by our system across such spam terms. In stark contrast from Figure 9 there is a very high percentage of splogs in the top 100 results.

Post Cleaning Evaluation

In BlogVox data cleaning improved results for opinion extraction. Figure 11 highlights the significance of identifying and removing extraneous content from blog posts. For 50 TREC queries, we fetched the first 500 matches from a Lucene index and used the baseline data cleaning heuristic. Some documents were selected only due to the presence of query terms in sidebars. Sometimes these are links to recent posts containing the query terms, but can often be links to advertisements, reading lists or link rolls, etc. Reducing the impact of sidebar on opinion rank through link elimination or feature weighing can improve search results.

Table III..3 shows the performance of the baseline heuristic and the SVM based data cleaner on a hand-tagged set of 400 links. The SVM model outperforms the baseline heuristic. The current data cleaning approach works by making a decision at the individual HTML tag level; we are currently working on automatically identifying the DOM subtrees that correspond to the sidebar elements.

Trec Submissions

The core BlogVox system produces results with two measures. The first is a relevance score ranging from 0.0 to 1.0, which is the value returned by the underlying Lucene query system. The second was a measure

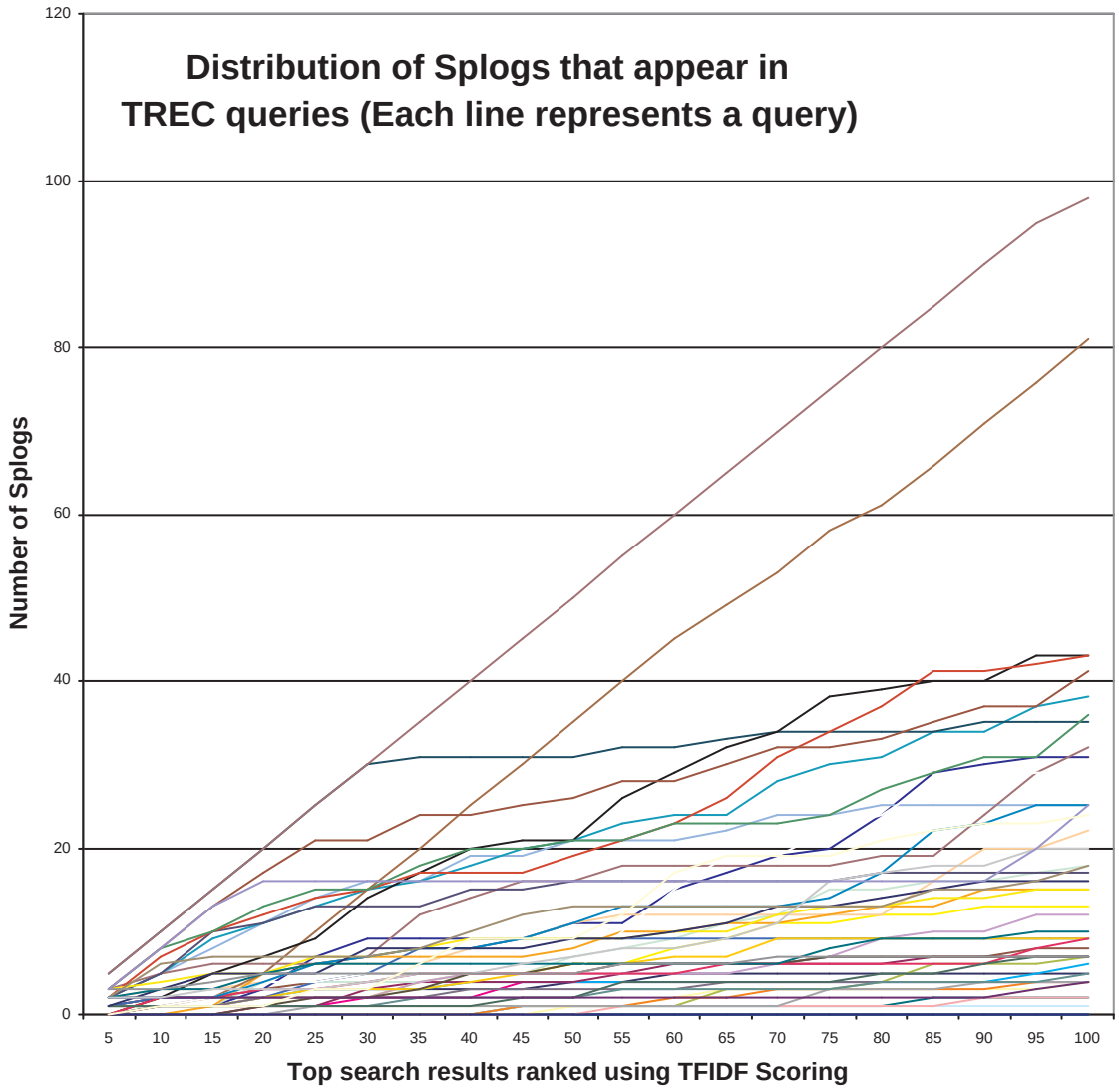


Figure 9: The number of splogs in the top x results for 50 TREC queries. Top splog queries include “cholesterol” and “hybrid cars”

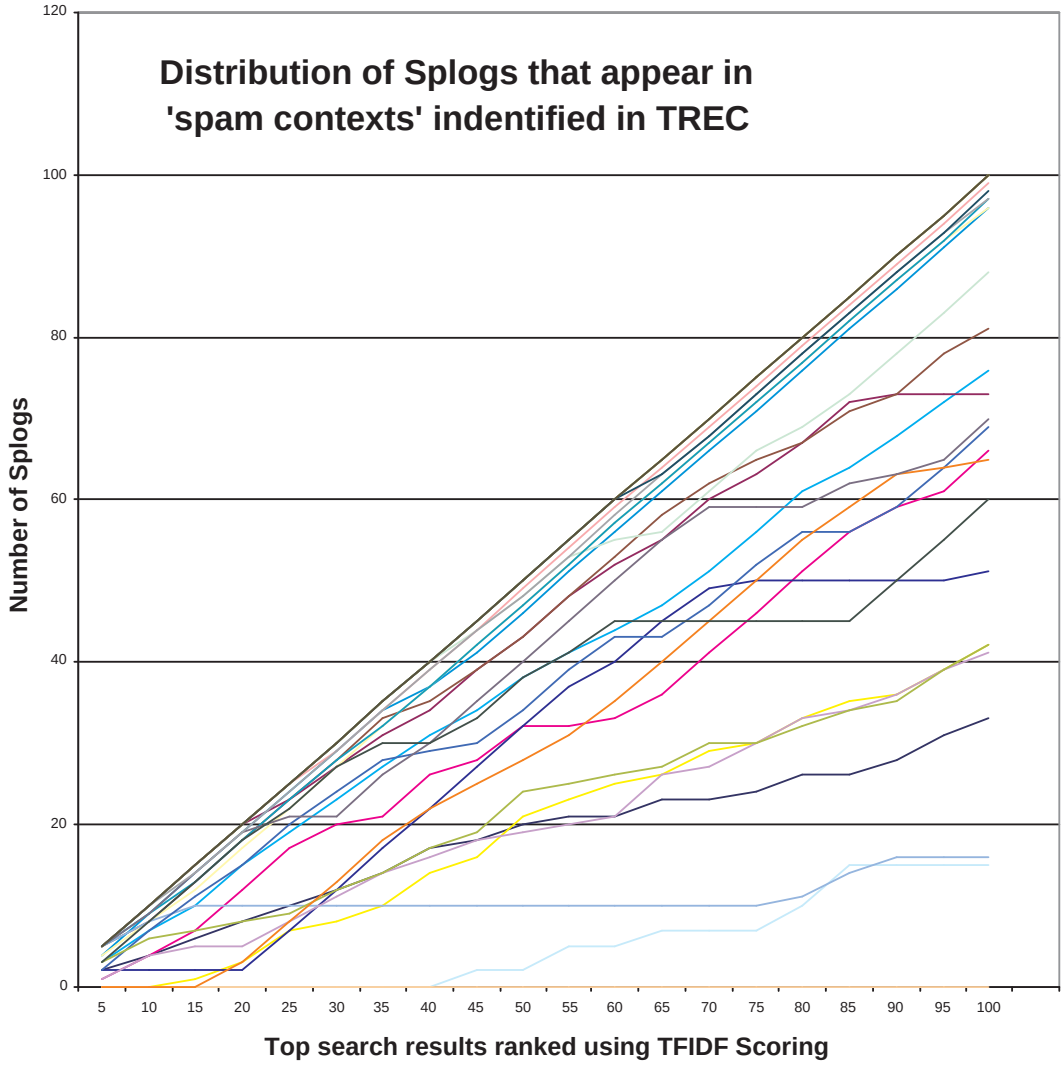


Figure 10: The number of splogs in the top x results of the TREC collection for 28 highly spammed query terms. Top splog queries include ’pregnancy’, ’insurance’, ’discount’

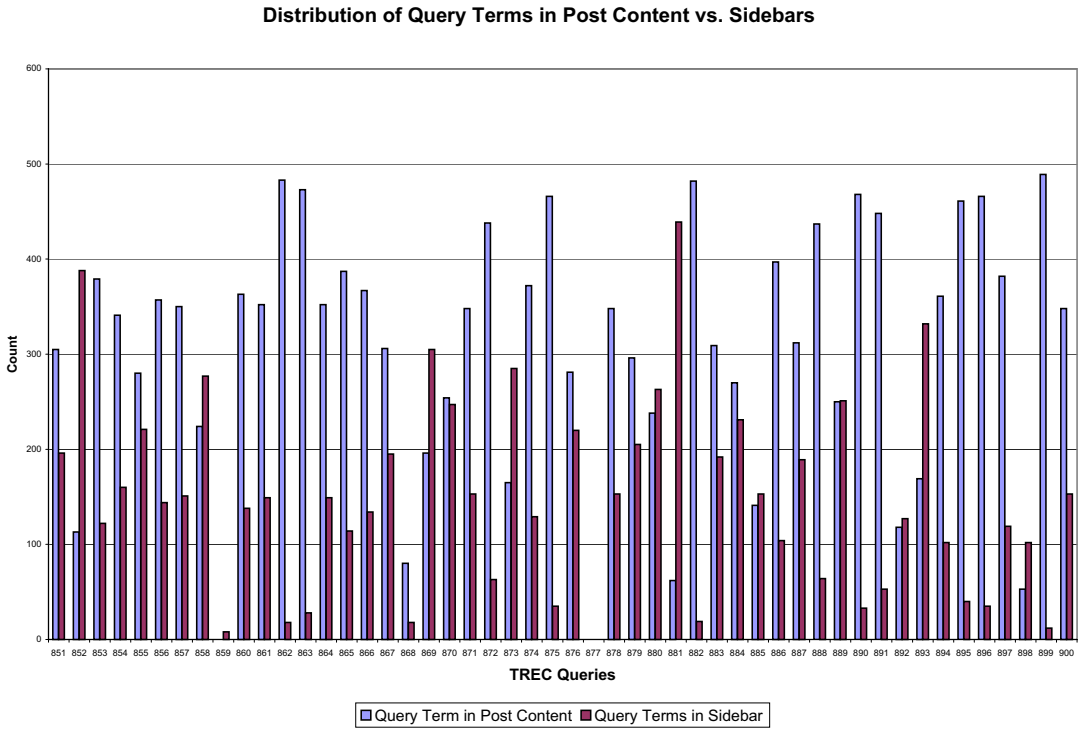


Figure 11: Documents containing query terms in the post title or content vs. exclusively in the sidebars, for 50 TREC queries, using 500 results fetched from the Lucene index.

run	opinion		topic relevance	
	map	r-prec	map	r-prec
UABas11	**0.0764**	**0.1307**	**0.1288**	**0.1805**
UAEx11	0.0586	0.0971	0.0994	0.1367
UAEx12	0.0582	0.0934	0.0985	0.1355
UAEx13	0.0581	0.0923	0.0978	0.1360
UAEx21	0.0590	0.0962	0.0998	0.1366
Corrected	**0.1275**	**0.202**	**0.1928**	**0.2858**
Cleaned	**0.1548**	**0.2388**	**0.2268**	**0.3272**

Table III..4: The results for the opinion and topic relevance performance of different runs

of opinionatedness, which was a real number greater than 0.0. We produced the sim numbers for each of the runs from a weighted average of the two numbers after normalizing them using the standard Z-normalization technique.

The baseline run was exectuted on the uncleaned dataset using a selection of what we anticipated to be the seven best scorer features and with an equal weighting for relevance and opinionatedness. This run was also the best performing run amongst our official runs. Runs two through five were made on the semi-cleaned dataset and using a larger set of eleven scorer features. After normalizing the result scores, we used weights of (1,1), (1,2), (1,3) and (2,1).

Figure 12 shows the results from the TREC submissions for opinion retrieval. Figure 13 shows the results for the topic relevance. The Mean Average Precision (MAP) for opinion retrieval of the original TREC submissions was 0.0764 and the R-Prec was around 0.1307. The MAP for topic relevance was about 0.1288 with an R-Prec of 0.1805. After inspection of the code, it appeared that this may have been due to a minor bug in the original code that was used for the official run. Upon correcting this and re-executing the run, we found that the MAP for opinion task was about 0.128 and for retrieval was about 0.1928. A final run was performed by running the queries against an index recreated by cleaning all the posts using heuristics described above. Table III..4 summarizes the results obtained. We find that cleaning significantly improved both opinion and retrieval scores of our system. Figure 15 compares the precision recall curves for these these runs.

We think that the retrieval performance could be improved by using the following approaches: use of query expansion modules, applying relevance feedback and using the description and narrative fields from the TREC queries to formulate the final Lucene query.

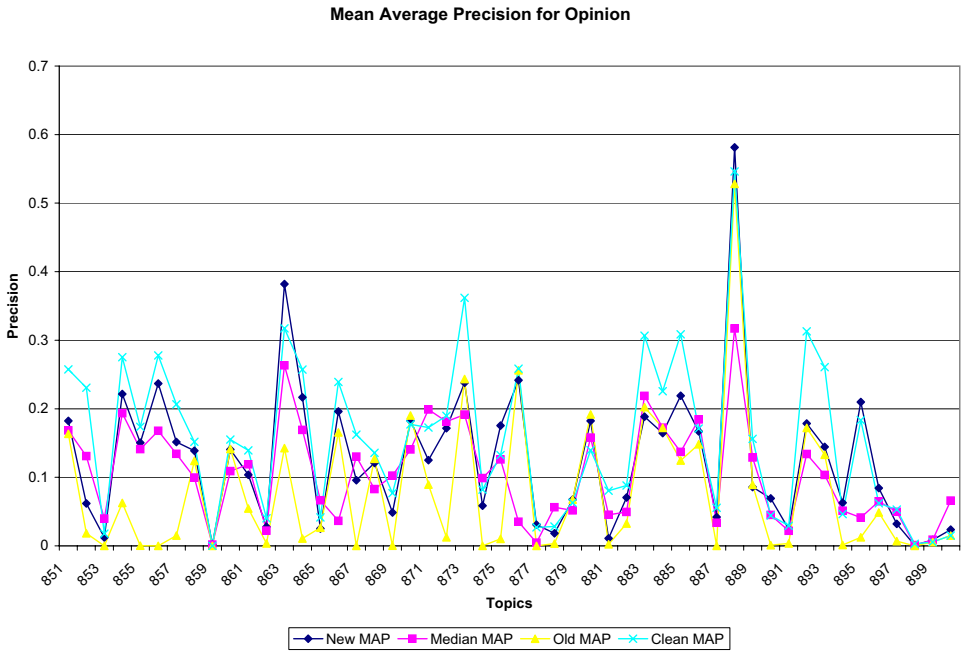


Figure 12: Mean average precision (for opinion) of original TREC submission UABas11 ,updated runs and clean index runs.

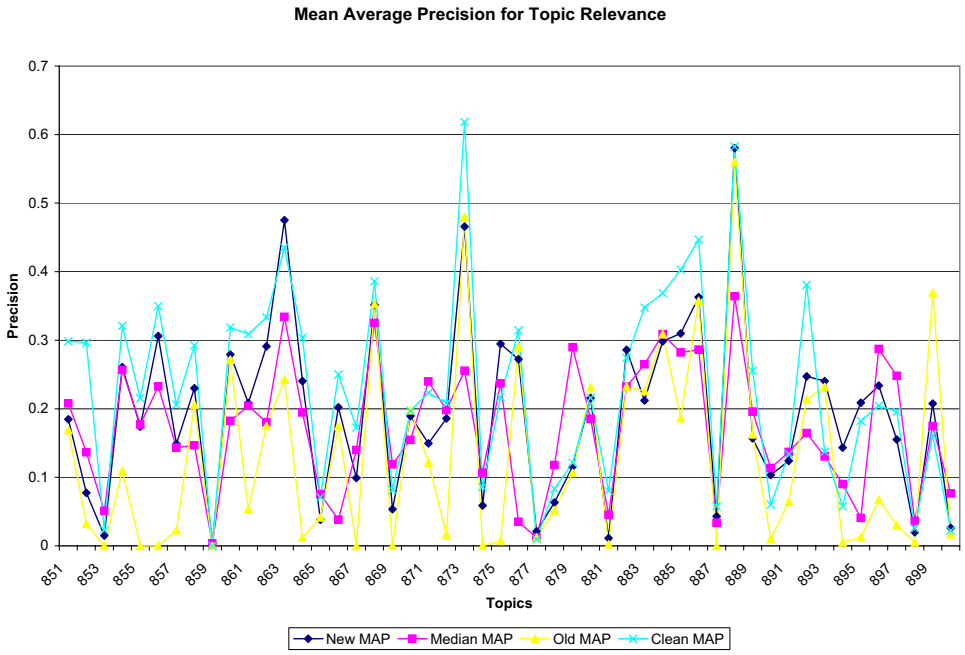


Figure 13: Mean average Precision (for topic relevance) of original TREC submission UABas11, updated runs and clean index runs.

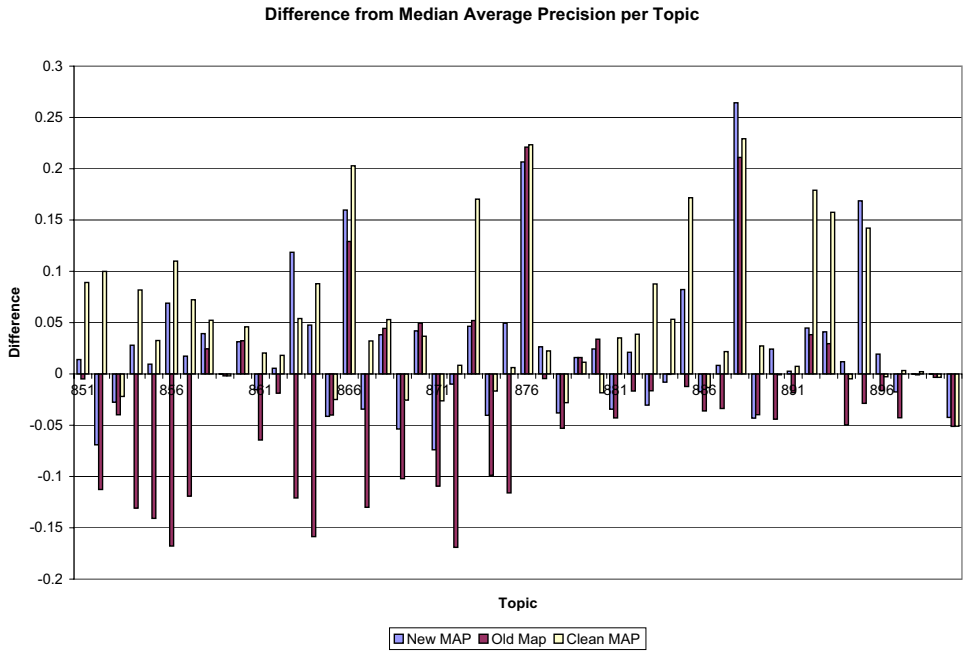


Figure 14: Difference of MAP from Median for original TREC submission UABas11, updated runs and clean index runs.

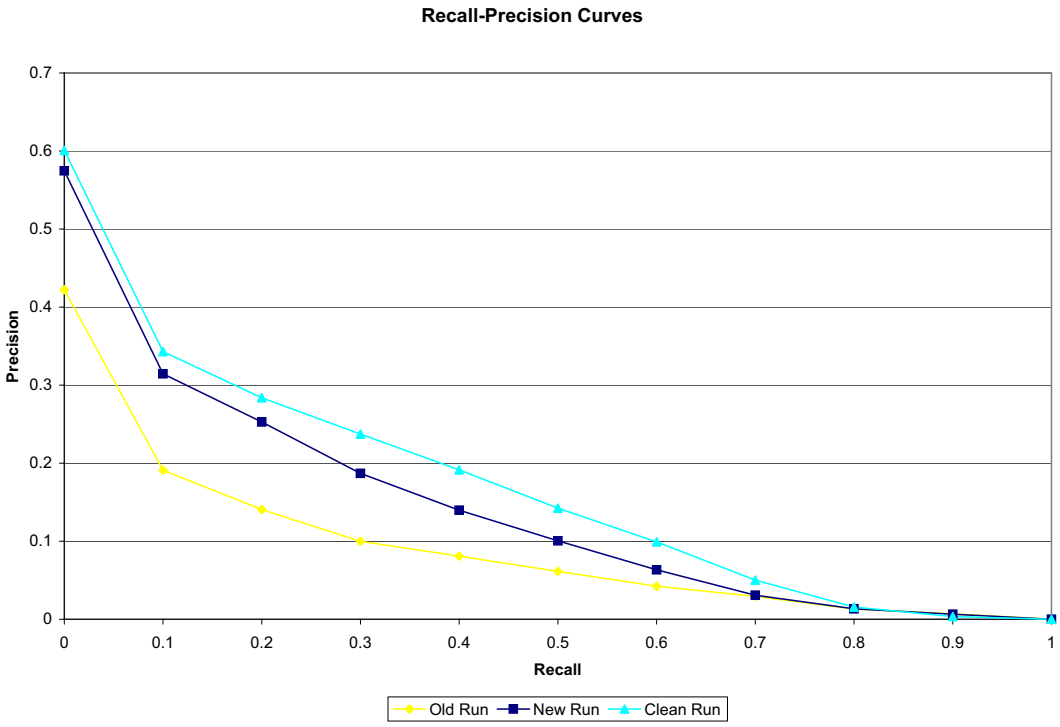


Figure 15: Precision Recall curves for original TREC submission UABas11, updated runs and clean index runs.

6. Conclusions

For TREC runs, we used an index on blog posts that had not been cleaned for all of the runs. For run one we evaluated these uncleaned posts using a complement of seven heuristics. For runs two through five, we retrieved a fixed number of post ids using the index of uncleaned data and then cleaned the resulting posts "on the fly". A larger set of eleven heuristic scoring functions was used for these runs. After cleaning a post, we did a heuristic check to ensure that at least some of the query terms remained. If not, the post was discarded. We believe that this ad hoc approach significantly lowered our precision scores for these runs due to at least three reasons. First, the relevance scores were computed by Lucene on the uncleaned posts and were not accurate for the cleaned versions since the term frequencies for both the collection and for each document were altered. Second, discarding many of the posts after the cleaning reduced the number of available results, already low due to the impending deadline. Finally, the cleaned posts were in many cases likely to be less relevant that their scores would indicate due to the removal of query words.

Manual inspection of some of the results showed that there were a number of matches that were due to the presence of the query terms in extraneous links. In order to verify the effectiveness of cleaning we created a new index using only the cleaned versions of the posts. We find that using this cleaner index improved not only retrieval results but also effective mean average precision for opinion retrieval. As can be observed from Figure 14, in almost all the cases the mean average precision for the runs on cleaned data outperform those on unclean data. The queries for which data cleaning made a significant improvement were "larry summers", "bruce bartlett", "Fox News Report" and "zyrtec". Comparing these with Figure 11 indicates that these were also queries that contained a higher number of matches that had the terms exclusively in the sidebar. On the other hand for queries like 'audi', 'oprah' and 'colbert report' the cleaned runs had a lower precision possibly due to the strict thresholds for cleaning.

We developed the BlogVox system as an opinion retrieval system for blog posts as part of the 2006 TREC Blog Track. This task requires processing an ad hoc queries representing topics and retrieving posts that express an opinion about them. Our initial experiments with the blog post collection revealed two problems: the presence of spam blogs and the large amounts of extra, non-content text in each posts.

We identified posts from spam blogs using a machine-learning based approach and eliminated them from the collection. The remaining posts were "cleaned" before being indexed to eliminate extraneous text associated with navigation links, blog-rolls, link-rolls, advertisements and sidebars. After retrieving posts relevant to a topic query, the system applies a set of scoring modules to each producing a vector of features estimating

the likelihood that a post expresses an opinion about the topic. These are combined using an SVM-based system and integrated with the overall relevancy score to rank the results.

Our evaluation of the BlogVox results showed that both splog elimination and post cleaning significantly increased the performance of the system. The overall performance as measured by the *mean average precision* and *R-precision* scores showed that the system worked well on most of the fifty test queries. We believe that the system can be improved by increasing the accuracy of the post-cleaning and refining the opinion scorers.

B. Semantic Analysis of RSS Feeds

The World Wide Web is a vast, open, accessible and free source of knowledge. With the rise of social media platforms the prefered means to publish and share information is via simple XML based syndication mechanisms supported by two popular protocols: RSS and ATOM. While these formats provide some structure to the documents, virtually most of the content in it and everywhere else on the Web is in natural language - a form difficult for most agents to directly understand. Many intelligent agents and applications on the Web need knowledge to support their reasoning and problem solving.

In this section, we describe our efforts in semantically enriching the content in RSS/ATOM feeds. We accomplish this by applying a matrure language understanding system on News feeds and publishing the output in the Semantic Web language OWL. This adds knowledge on the Web in a form designed for agents to consume easily. SemNews is a prototype system that monitors a number of RSS news feeds. As new stories are published the system extracts the relevant meta-data associated with it and indexes the articles. The news snippets or summaries provided in the RSS streams are then semantically analyzed using OntoSem, an ontology based natural language procesing system. OntoSem performs a syntactic parse of the sentences followed by a semantic and pragmatic analysis of the text. SemNews then publishes and exports the interpreted meaning representation in OWL. Additionally, it also stores all the TMRs in a triple store. In addition the underlying ontology is also exported to OWL.

By semanticizing RSS news streams, one can now explore the stories via concepts defined in an ontology. Moreover, SemNews also provides a way to perform structured queries (via RDQL/SPARQL[5]) over the text meaning representation of natural language text. This allows users to search specific news article using high level concepts like: "*Find all the stories in which a politician visited Middle East*", which would match a news item that talks about the defense secretary, Condoleezza Rice visiting Iraq. Semantic alerts can be set, where an new RSS feed is generated for each of the RDQL/SPARQL queries so that whenever a new story matches the query the user is notified.

1. Related Work

Among past projects that have addressed semantic annotation are the following:

- Gildea and Jurafsky [53] created a stochastic system that labels case roles of predicates with either abstract (e.g., AGENT, THEME) or domain-specific (e.g., MESSAGE, TOPIC) roles. The system

[5]RDF Data Query Language; SPARQL Protocol and RDF Query Language

trained on 50,000 words of hand-annotated text (produced by the FrameNet project). When tasked to segment constituents and identify their semantic roles (with fillers being undisambiguated textual strings, not machine-tractable instances of ontological concepts, as in OntoSem), the system scored in the 60s in precision and recall. Limitations of the system include its reliance on hand-annotated data, and its reliance on prior knowledge of the predicate frame type (i.e., it lacks the capacity to disambiguate productively). Semantics in this project is limited to case-roles.

- The goal of the Interlingual Annotation of Multilingual Text Corpora project [6] is to create a syntactic and semantic annotation representation methodology and test it out on six languages (English, Spanish, French, Arabic, Japanese, Korean, and Hindi). The semantic representation, however, is restricted to those aspects of syntax and semantics that developers believe can be consistently handled well by hand annotators for many languages. The current stage of development includes only syntax and light semantics essentially, thematic roles.

- In the ACE project [7], annotators carry out manual semantic annotation of texts in English, Chinese and Arabic to create training and test data for research task evaluations. The downside of this effort is that the inventory of semantic entities, relations and events is very small and therefore the resulting semantic representations are coarse-grained: e.g., there are only five event types. The project description promises more fine-grained descriptors and relations among events in the future. Another response to the clear insufficiency of syntax-only tagging is offered by the developers of PropBank, the Penn Treebank semantic extension. Kingsbury et al. [104] report: It was agreed that the highest priority, and the most feasible type of semantic annotation, is coreference and predicate argument structure for verbs, participial modifiers and nominalizations, and this is what is included in PropBank.

Recently, there has been a lot of interest in applying Information extraction technologies for the Semantic Web. However, few systems capable of deeper semantic analysis have been applied in Semantic Web related tasks. Information extraction tools work best when the types of objects that need to be identified are clearly defined, for example the objective in MUC [63] was to find the various named entities in text. Using OntoSem, we aim to not only provide such information, but also convert the text meaning representation of natural language sentences into Semantic Web representations.

[6] http://aitc.aitcnet.org/nsf/iamtc/
[7] http://www.ldc.upenn.edu/Projects/ACE/intro.html

A project closely related to our work was an effort to map the Mikrokosmos knowledge base to OWL [12, 13]. Mikrokosmos is a precursor to OntoSem and was developed with the original idea of using it as an interlingua in machine translation related work. This project developed some basic mapping functions that can create the class hierarchy and specify the properties and their respective domains and ranges. In our system we describe how facets, numeric attribute ranges can be handled and more importantly we describe a technique for translating the sentences from their Text Meaning Representation to the corresponding OWL representation thereby providing semantically marked up Natural Language text for use by other agents.

Oliver et al. [36] describe an approach to representing the Foundational Model of Anatomy (FMA) in OWL. FMA is a large ontology of the human anatomy and is represented in a frame-based knowledge representation language. Some of the challenges faced were the lack of equivalent OWL representations for some frame based constructs and scalability and computational issues with the current reasoners.

Schlangen et al. [180] describe a system that combines a natural language processing system with Semantic Web technologies to support the content-based storage and retrieval of medical pathology reports. The NLP component was augmented with a background knowledge component consisting of a a domain ontology represented in OWL. The result supported the extraction of domain specific information from natural language reports which was then mapped back into a Semantic Web representation.

TAP [176] is an open source project lead by Stanford University and IBM Research aimed at populating the Semantic Web with information by providing tools that make the web a giant distributed Database. TAP provides a set of protocols and conventions that create a coherent whole of independently produced bits of information, and a simple API to navigate the graph. Local, independently managed knowledge bases can be aggregated to form selected centers of knowledge useful for particular applications.

Kruger et al. [115] developed an application that learned to extract information from talk announcements from training data using an algorithm based on Stalker [150]. The extracted information was then encoded as markup in the Semantic Web language DAML+OIL, a precursor to OWL. The results were used as part of the ITTALKS system [33].

The Haystack Project has developed system [78] enabling users to train a browsers to extract Semantic Web content from HTML documents on the Web. Users provide examples of semantic content by highlighting them in their browser and then describing their meaning. Generalized wrappers are then constructed to extract information and encode the results in RDF. The goal is to let individual users generate Semantic Web content from text on web pages of interest to them.

The On-to-Knowledge project [35] provides an ontology-based system for knowledge management. It uses Ontology-based Inference Layer (OIL) to support for description logics (DL) and frame-based systems over the WWW. OWL itself is an extension derived from OIL and DAML. The OntoExtract and OntoWrapper sub-system in On-to-knowledge were responsible for processing unstructured and structured text. These systems were used to automatically extract ontologies and express them in Semantic Web representations. At the heat of OntoExtract is an NLP system that process text to perform lexical and semantic analysis. Finally, concepts found in free text are represented as an ontology.

The Cyc project has developed a very large knowledge base of common sense facts and reasoning capabilities. Recent efforts [205] include the development of tools for automatically annotating documents and exporting the knowledge in OWL. The authors also highlight the difficulties in exporting an expressive representation like CycL into OWL due to lack of equivalent constructs.

2. OntoSem

Ontological Semantics (OntoSem) is a theory of meaning in natural language text [162]. The OntoSem environment is a rich and extensive tool for extracting and representing meaning in a language independent way. The OntoSem system is used for a number of applications such as machine translation, question answering, information extraction and language generation. It is supported by a *constructed world model* encoded as a rich ontology. The Ontology is represented as a directed acyclic graph using IS-A relations. It contains about 8000 concepts that have on an average 16 properties per concept. At the topmost level the concepts are: OBJECT, EVENT and PROPERTY.

The OntoSem ontology is expressed in a frame-based representation and each of the frames corresponds to a concept. The concepts are defined using a collection of slots that could be linked using IS-A relations. A slot consists of a PROPERTY, FACET and a FILLER.

```
ONTOLOGY ::= CONCEPT+
CONCEPT  ::= ROOT | OBJECT-OR-EVENT | PROPERTY
SLOT     ::= PROPERTY + FACET + FILLER
```

A property can be either an attribute, relation or ontology slot. An ontology slot is a special type of property that is used to describe and organize the ontology. The ontology is closely tied to the lexicon to make it language independent. There is a lexicon for each language and stored “meaning procedures” that are used to disambiguate word senses and references. Thus keeping the concepts defined relatively few and

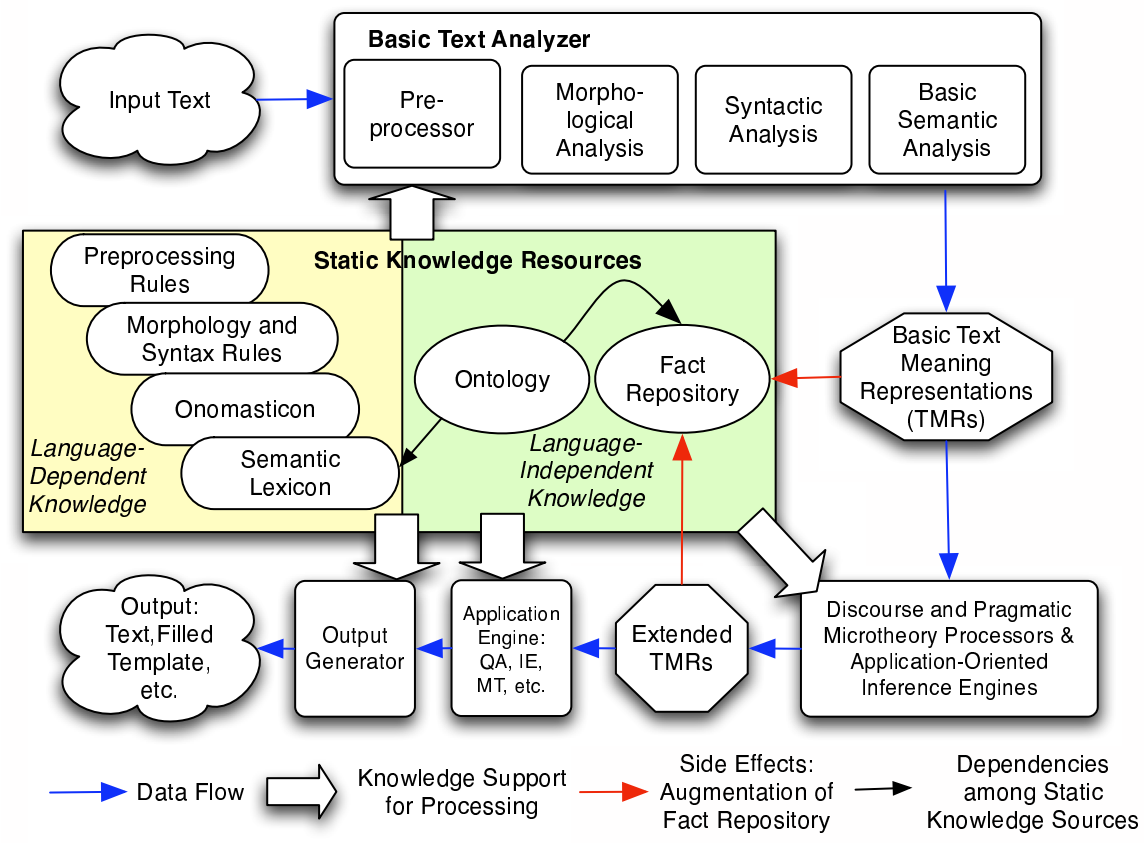


Figure 16: High level view of OntoSem

making the ontology small. Text analysis relies on extensive static knowledge resources, some of which are described below:

- The OntoSem language-independent ontology, which currently contains around 8,500 concepts, each of which is described by an average of 16 properties. The ontology is populated by concepts that we expect to be relevant cross-linguistically. The current experiment was run on a subset of the ontology containing about 6,000 concepts.

- An OntoSem lexicon whose entries contain syntactic and semantic information (linked through variables) as well as calls for procedural semantic routines when necessary. The semantic zone of an entry most frequently refers to ontological concepts, either directly or with property-based modifications, but can also describe word meaning extra-ontologically, for example, in terms of modality, aspect or time (see McShane and Nirenburg 2005 for in-depth discussion of the lexicon/ontology connection). The current English lexicon contains approximately 30,000 senses, including most closed-class items and many of the most frequent and polysemous verbs, as selected through corpus analysis. The base lexicon is expanded at runtime using an inventory of lexical (e.g., derivational-morphological) rules.

- An onomasticon, or lexicon of proper names, which contains approximately 350,000 entries.

- A fact repository, which contains remembered instances of ontological concepts (e.g., SPEECH-ACT-3366 is the 3366th instantiation of the concept SPEECH-ACT in the memory of a text-processing agent). The fact repository is not used in the current experiment but will provide valuable semantically-annotated context information for future experiments.

- The OntoSem syntactic-semantic analyzer, which performs preprocessing (tokenization, named-entity and acronym recognition, etc.), morphological, syntactic and semantic analysis, and the creation of TMRs.

- The TMR language, which is the metalanguage for representing text meaning.

OntoSem knowledge resources have been acquired by trained acquirers using a broad variety of efficiency-enhancing tools graphical editors, enhanced search facilities, capabilities of automatically acquiring knowledge for classes of entities on the basis of manually acquired knowledge for a single representative of the class, etc. OntoSems DEKADE environment [138] facilitates both knowledge acquisition and semi-automatic creation of gold standard TMRs, which can be also viewed as deep semantic text annotation.

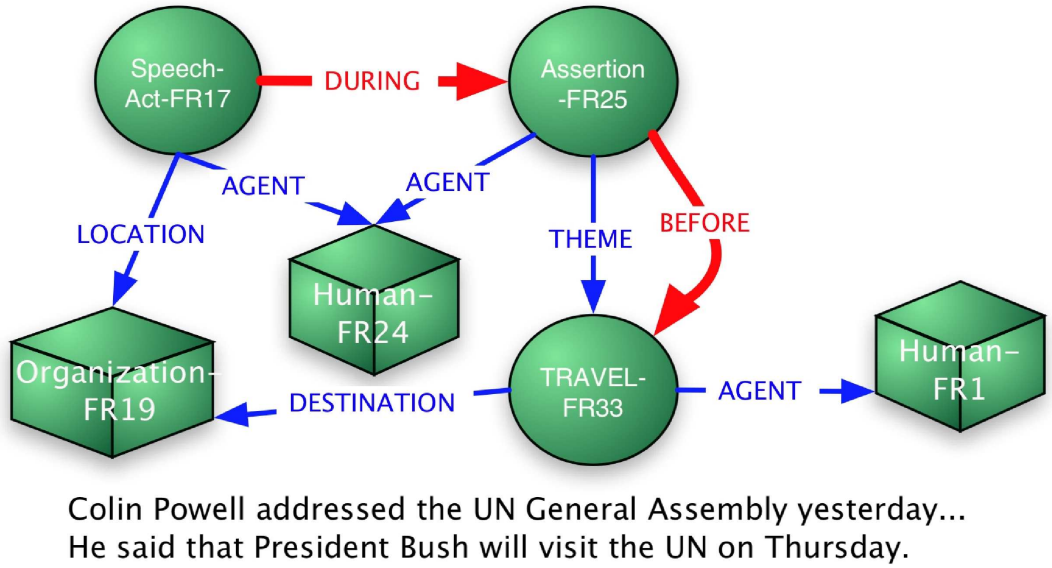


Figure 17: OntoSem goes through several basic stages in converting a sentence into a text meaning representation (TMR).

The OntoSem environment takes as input unrestricted text and performs different syntactic and semantic processing steps to convert it into a set of Text Meaning Representations (TMR). The basic steps in processing the sentence to extract the meaning representation is show in figure 16. The preprocessor deals with identifying sentence and word boundaries, part of speech tagging, recognition of named entities and dates, etc. The syntactic analysis phase identifies the various clause level dependencies and grammatical constructs of the sentence. The TMR is a representation of the meaning of the text and is expressed using the various concepts defined in the ontology. The TMRs are produced as a result of semantic analysis which uses knowledge sources such as lexicon, onomasticon and fact repository to resolve ambiguities and time references. TMRs have been used as the substrate for question-answering [10], machine translation [11] and knowledge extraction. Once the TMRs are generated, OntoSem2OWL converts them to an equivalent OWL representation.

The learned instances from the text are stored in a *fact repository* which essentially forms the knowledge base of OntoSem. As an example the sentence: *" He (Colin Powell) asked the UN to authorize the war"* is converted to the TMR shown in Figure 18. A more detailed description of OntoSem and its features is available in [163] and [1].

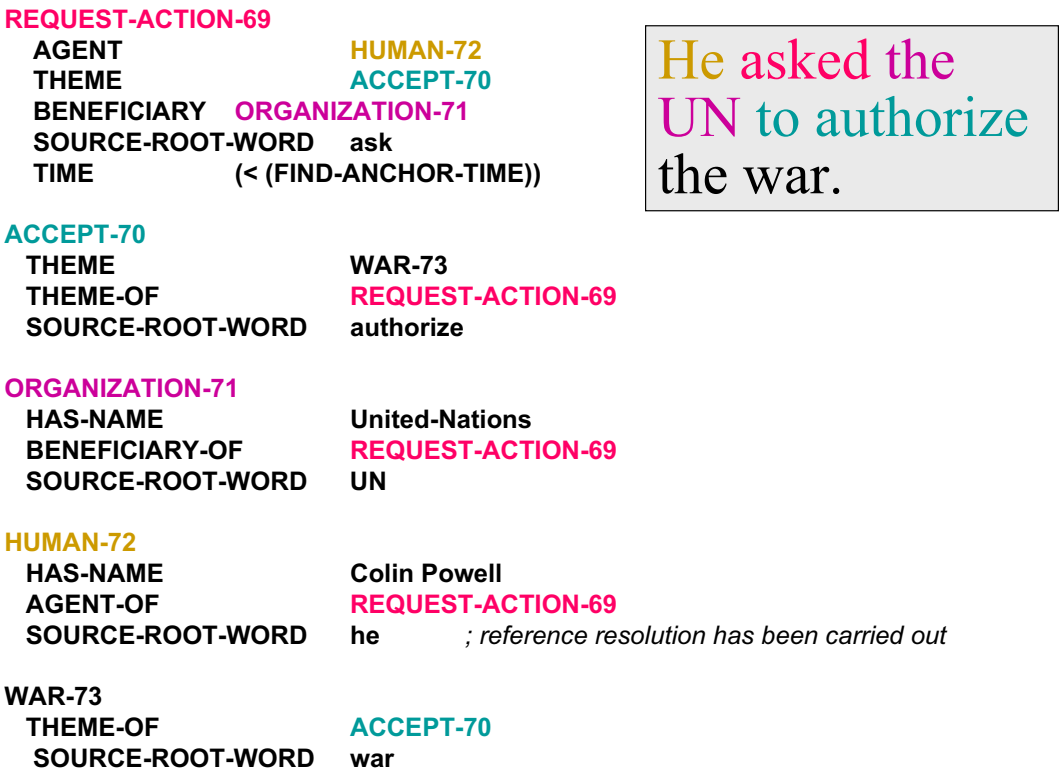


Figure 18: OntoSem constructs this text meaning representation (TMR) for the sentence *"He (Colin Powell) asked the UN to authorize the war"*.

3. Making RSS Machine Readable

We have developed **OntoSem2OWL** [85] as a tool to convert OntoSem's ontology and TMRs encoded in it to OWL. This enables an agent to use OntoSem's environment to extract semantic information from natural language text. Ontology Mapping deals with defining functions that describe how concepts in one ontology are related to the concepts in some other ontology [39]. Ontology translation process converts the sentences that use the source ontology into their corresponding representations in the target ontology. In converting the OntoSem Ontology to OWL, we are performing the following tasks:

- Translating the OntoSem ontology deals with mapping the semantics of OntoSem into a corresponding OWL version.
- Once the ontology is translated the sentences that use the ontology are syntactically converted.
- In addition OntoSem is also supported by a fact repository which is also mapped to OWL.

OntoSem2OWL is a rule based translation engine that takes the OntoSem Ontology in its LISP representation and converts it into its corresponding OWL format. The following is an example of how a concept ONTOLOGY-SLOT is described in OntoSem:

```
(make-frame definition
  (is-a (value (common ontology-slot)))
  (definition (value (common "Human
       readable explanation for a concept")))
  (domain (sem (common all))))
```

Its corresponding OWL representation is:

```
<owl:ObjectProperty rdf:ID="definition">
 <rdfs:subPropertyOf>
   <owl:ObjectProperty rdf:about="#ontology-slot"/>
 </rdfs:subPropertyOf>
 <rdfs:label>
   "Human readable explanation for a concept"
 </rdfs:label>
```

	case	times used	mapped using
1	total Class/Property make-frame	8199	owl:class or owl:ObjectProperty
2	Definition	8192	rdfs:label
3	is-a relationship	8189	owl:subClassOf

Table III..5: Table showing how often each of the Class related constructs are used

```
  <rdfs:domain>
   <owl:Class rdf:about="#all"/>
  </rdfs:domain>
</owl:ObjectProperty>
```

We will briefly describe how each of the OntoSem features are mapped into their OWL versions: classes, properties, facets, attribute ranges and TMRs.

Handling Classes

New concepts are defined in OntoSem using *make-frame* and related to other concepts using the *is-a* relation. Each concept may also have a corresponding definition. Whenever the system encounters a *make-frame* it recognizes that this is a new concept being defined. OBJECT or EVENT are mapped to *owl:Class* while, PROPERTIES are mapped to *owl:ObjectProperty*. ONTOLOGY-SLOTS are special properties that are used to structure the ontology. These are also mapped to *owl:ObjectProperty*. Object definitions are created using *owl:Class* and the IS-A relation is mapped using *owl:subClassOf*. Definition property in OntoSem has the same function as *rdfs:label* and is mapped directly. The table III..5 shows the usage of each of these features in OntoSem.

Handling Properties

Whenever the level 1 parent of a concept is of the type PROPERTY it is translated to *owl:ObjectProperty*. Properties can also be linked to other properties using the IS-A relation. In case of properties, the IS-A relation maps to the *owl:subPropertyOf*. Most of the properties also contain the domain and the range slots. Domain defines the concepts to which the property can be applied and the ranges are the concepts that the property slot of an instance can have as fillers. OntoSem domains are converted to *rdfs:domain* and ranges are converted to *rdfs:range*. For some of the properties OntoSem also defines inverses using the INVERSE-OF relationship. It can be directly mapped to the *owl:inverseOf* relation.

In case there are multiple concepts defined for a particular domain or range, OntoSem2OWL handles it using *owl:unionOf* feature. For example:

```
(make-frame controls
 (domain
  (sem (common physical-event
               physical-object
               social-event
               social-role)))
  (range (sem (common actualize
                      artifact
                      natural-object
                      social-role)))
  (is-a (value (common relation)))
  (inverse (value (common controlled-by)))
  (definition
    (value (common
     "A relation which relates concepts to
      what they can control"))))
```

is mapped to

```
<owl:ObjectProperty rdf:ID= "controls">
 <rdfs:domain>
  <owl:Class>
    <owl:unionOf rdf:parseType="Collection">
      <owl:Class rdf:about="#physical-event"/>
      <owl:Class rdf:about="#physical-object"/>
      <owl:Class rdf:about="#social-event"/>
      <owl:Class rdf:about="#social-role"/>
   </owl:unionOf>
  </owl:Class>
```

	case	frequency	mapped using
1	domain	617	rdfs:domain
2	domain with not facet	16	owl:disjointWith
3	range	406	rdfs:range
4	range with not facet	5	owl:disjointWith
5	inverse	260	owl:inverseOf

Table III..6: Table showing how often each of the Property related constructs are used

```
 </rdfs:domain>
 <rdfs:range>
  <owl:Class>
   <owl:unionOf rdf:parseType="Collection">
      <owl:Class rdf:about="#actualize"/>
<owl:Class rdf:about="#artifact"/>
<owl:Class rdf:about="#natural-object"/>
<owl:Class rdf:about="#social-role"/>
   </owl:unionOf>
  </owl:Class>
 </rdfs:range>
 <rdfs:subPropertyOf>
   <owl:ObjectProperty rdf:about="#relation"/>
 </rdfs:subPropertyOf>
 <owl:inverseOf rdf:resource="#controlled-by"/>
 <rdfs:label>
  "A relation which relates concepts to
   what they can control"
 </rdfs:label>
</owl:ObjectProperty>
```

The table III..6 describes the typical usages of the property related constructs in OntoSem.

Handling Facets

OntoSem uses facets as a way of restricting the fillers that can be used for a particular slot. In OntoSem there are six facets that are created and one,*inv* that is automatically generated. The table III..7 shows the different facets and how often they are used in OntoSem.

- *SEM and VALUE*: These are the most commonly used facets. OntoSem2OWL handles these identically and are maps them using *owl:Restriction* on a particular property. Using *owl:Restriction* we can locally restrict the type of values a property can take unlike *rdfs:domain* or *rdfs:range* which specifies how the property is globally restricted [137].

- *RELAXABLE-TO*: This facet indicates that the value for the filler can take a certain type. It is a way of specifying “typical violations”. One way of handling RELAXABLE-TO is to add this information in an annotation and also add this to the classes present in the *owl:Restriction*.

- *DEFAULT*: OWL provides no clear way of representing defaults, since it only supports monotonic reasoning and this is one of the issues that have been expressed for future extensions of OWL language [81]. These issues need to be further investigated in order to come up with an appropriate equivalent representation in OWL. One approach is to use rule languages like SWRL [80] to express such defaults and exceptions. Another approach would be to elevate facets to properties. This can be done by combining the property-facet to make a new property. Thus a concept of an apple that has a property color with the default facet value ’red’ could be translated to a new property in the owl version of the frame where the property name is color-default and it can have a value of red.

- *DEFAULT-MEASURE*: This facet indicates what the typical units of measurements are for a particular property. This can be handled by creating a new property named MEASURING-UNITS or adding this information as a rule.

- *NOT*: This facet specifies that certain values are not permitted in the filler of the slot in which this is defined. *NOT* facet can be handled using the *owl:disjointWith* feature.

- *INV*: This facet need not be handled since this information is already covered using the inverse property which is mapped to *owl:inverseOf*.

Although DEFAULT and DEFAULT-MEASURE provides useful information, it can be noticed from III..7 that relatively they are used less frequently. Hence in our use cases, ignoring these facets does not lose a lot

	case	frequency	mapped using
1	value	18217	owl:Restriction
2	sem	5686	owl:Restriction
3	relaxable-to	95	annotation
4	default	350	**not handled**
5	default-measure	612	**not handled**
6	not	134	owl:disjointWith
7	inv	1941	not required

Table III..7: Table showing how often each of the facets are used

of information.

Handling Attribute Ranges

Certain fillers can also take numerical ranges as values. For instance the property *age* can take a numerical value between 0 and 120 for instance. Additionally $<, >, <>$ could also be used in TMRs. Attribute ranges can be handled using XML Schema [3] in OWL. The following is an example of how the property *age* could be represented in OWL using *xsd:restriction*:

```
<xsd:restriction base="integer">
   <xsd:minInclusive value="0">
   <xsd:maxExclusive value="120">
</xsd:restriction>
```

Converting Text Meaning Representations

Once the OntoSem ontology is converted into its corresponding OWL representation, we can now translate the text meaning representations into statements in OWL. In order to do this we can use the namespace defined as the OntoSem ontology and use the corresponding concepts to create the representation. The TMRs also contain additional information such as ROOT-WORDS and MODALITY. These are used to provide additional details about the TMRs and are added to the annotations. In addition TMRs also contain certain triggers for 'meaning procedures' such as TRIGGER-REFERENCE and SEEK-SPECIFICATION. These are actually procedural attachments and hence can not be directly mapped into the corresponding OWL versions.

Sentence: *Ohio Congressman Arrives in Jordan*

TMR

```
(COME-1740
```

```
 (TIME (VALUE (COMMON (FIND-ANCHOR-TIME))))
 (DESTINATION (VALUE (COMMON CITY-1740)))
 (AGENT (VALUE (COMMON POLITICIAN-1740)))
 (ROOT-WORDS (VALUE (COMMON (ARRIVE))))
 (WORD-NUM (VALUE (COMMON 2)))
 (INSTANCE-OF (VALUE (COMMON COME))))
```

TMR in OWL

```
<ontosem:come rdf:about="COME-1740">
 <ontosem:destination
          rdf:resource="#CITY-1740"/>
 <ontosem:agent
    rdf:resource="#POLITICIAN-1740"/>
</ontosem:come>
```

TMR

```
(POLITICIAN-1740
  (AGENT-OF (VALUE (COMMON COME-1740)))
  ;; Politician with some relation to Ohio. A
  ;; later meaning procedure should try to find
  ;; that the relation is that he lives there.
  (RELATION (VALUE (COMMON PROVINCE-1740)))
  (MEMBER-OF (VALUE (COMMON CONGRESS)))
  (ROOT-WORDS (VALUE (COMMON (CONGRESSMAN))))
  (WORD-NUM (VALUE (COMMON 1)))
  (INSTANCE-OF (VALUE (COMMON POLITICIAN))))
```

TMR in OWL

```
<ontosem:politician rdf:about="POLITICIAN-1740">
  <ontosem:agent-of rdf:resource="#COME-140"/>
  <ontosem:relation rdf:resource="#PROVINCE-1740"/>
```

```
  <ontosem:member-of rdf:resource="#congress"/>
</ontosem:politician>
```

TMR

```
(CITY-1740
  (HAS-NAME (VALUE (COMMON "JORDAN")))
  (ROOT-WORDS (VALUE (COMMON (JORDAN))))
  (WORD-NUM (VALUE (COMMON 4)))
  (DESTINATION-OF (VALUE (COMMON COME-1740)))
  (INSTANCE-OF (VALUE (COMMON CITY))))
```

TMR in OWL

```
<ontosem:city rdf:about="CITY-1740">
  <ontosem:has-name>JORDAN</ontosem:has-name>
  <ontosem:destination-of rdf:resource="#COME-1740"/>
</ontosem:city>
```

4. Semantic News Framework

One of the motivations for integrating language understanding agents into the Semantic Web is to enable applications to use the information published in free text along with other Semantic Web data. SemNews [8] [86] is a semantic news service that monitors different RSS news feeds and provides structured representations of the meaning of news articles found in them. As new articles appear, SemNews extracts the summary from the RSS description and processes it with OntoSem. The resulting TMR is then converted into OWL. This enables us to *semantacize* the RSS content and provide live and up-to-date content on the Semantic Web. The prototype application also provides a number of interfaces which allow users and agents to query over the meaning representation of the text as expressed in OWL.

Figure 19 shows the basic architecture of SemNews. The RSS feeds from different news sources are aggregated and parsed. These RSS feeds are also rich in useful meta-data such as information on the author, the date when the article was published, the news category and tag information. These form the explicit meta-data that is provided by the publisher. However there is a large portion of the RSS field that is essentially plain

[8] http://semnews.umbc.edu

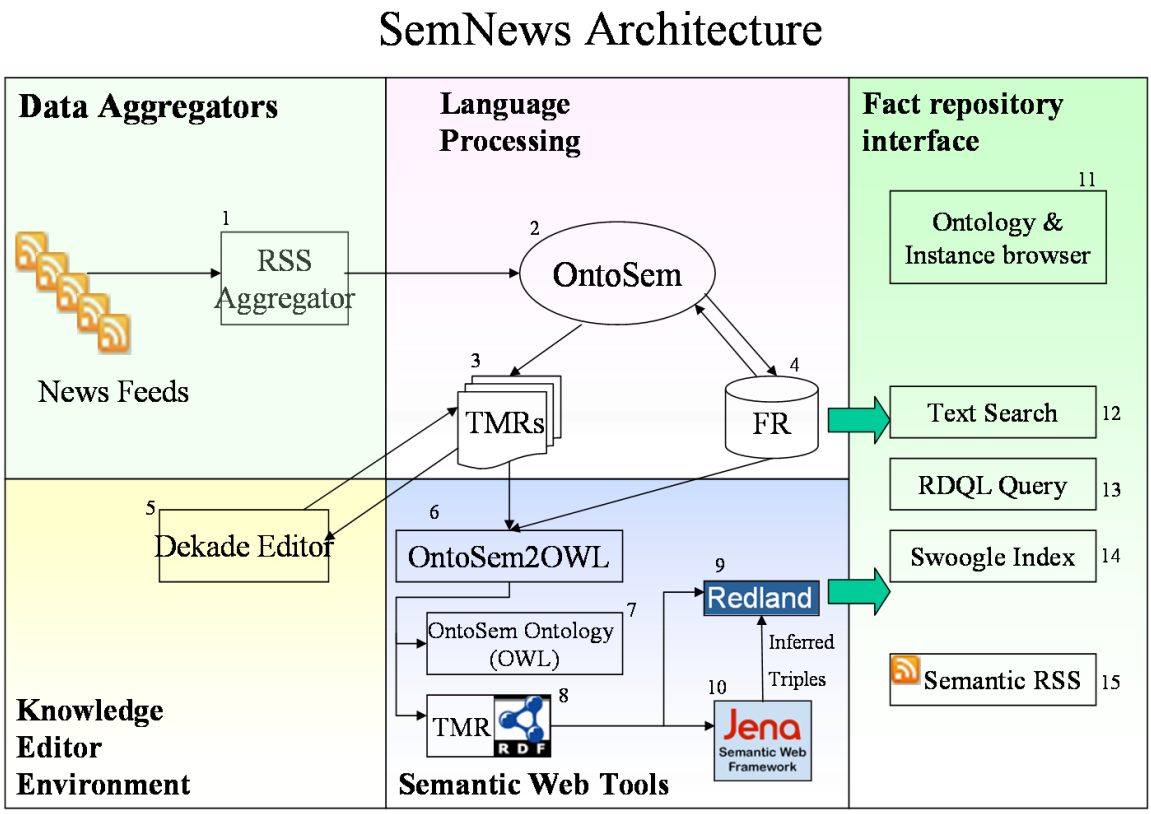


Figure 19: The SemNews application, which serves as a testbed for our work, has a simple architecture. RSS (1) from multiple sources is aggregated and then processed by the OntoSem (2) text processing environment. This results in the generation of TMRs (3) and updates to the fact repository (4). The Dekade environment (5) can be used to edit the ontology and TMRs. OntoSem2OWL (6) converts the ontology and TMRs to their corresponding OWL versions (7,8). The TMRs are stored in the Redland triple store (9) and additional triples inferred by Jena (10). There are also multiple viewers for searching and browsing the fact repository and triple store.

text and does not contain any semantics in them. It would be of great value if this text available in description and comment fields for example could be *semantacized*. By using Natural Language Processing (NLP) tools such as OntoSem we can convert natural language text into a structured representation thereby adding additional metadata in the RSS fields. Once processed, it is converted to its Text Meaning Representation (TMR). OntoSem also updates its fact repositories to store the information found in the sentences processed. These facts extracted help the system in its future text analysis tasks.

An optional step of correction of the TMRs could be performed by means of the Dekade environment [49]. This is helpful in correcting cases where the analyzers are not able to correctly annotate parts of the sentence. Corrections can be performed at both the syntactic processor and the semantic analyzer phase. The Dekade environment could also be used to edit the OntoSem ontology and lexicons or static knowledge sources.

As discussed in the previous sections, the meaning in these structured representations, also known as Text Meaning Representations (TMR), can be preserved by mapping them to OWL/RDF. The OWL version of a document's TMRs is stored in a Redland-based triple store, allowing other applications and users to perform semantic queries over the documents. This enables them to search for information that would otherwise not be easy to find using simple keyword based search. The TMRs are also indexed by the Swoogle Semantic Web Search system [43].

The following are some examples of queries that go beyond simple keyword searches.

- **Conceptually searching for content.** Consider the query *"Find all stories that have something to do with a place and a terrorist activity"*. Here the goal is to find the content or the story, but essentially by means of using ontological concepts rather than string literals. So for example, since we are using the ontological concepts here, we could actually benefit from resolving different kinds of terror events such as bombing or hijacking to a terrorist-activity concept.

- **Context based querying.** Answering the query *"Find all the events in which 'George Bush' was a speaker"* involves finding the context and relation in which a particular concept occurs. Using named entity recognition alone, one can only find that there is a story about a named entity of the type person/human, however it is not directly perceivable as to what role the entity participated in. Since OntoSem uses deeper semantics, it not only identifies the various entities but also extracts the relations in which these entities or instances participate, thereby providing additional contextual information.

- **Reporting facts.** To answer a query like *"Find all politicians who traveled to 'Asia'"* requires reasoning about people's roles and geography. Since we are using ontological concepts rather than plain text and we have certain relations like meronomy/part-of we could recognize that Colin Powel's trip to China will yield an answer.

- **Knowledge sharing on the semantic web.** Knowledge sharing is critical for agents to reason on the semantic web. Knowledge can be shared by means of using a common ontology or by defining mappings between existing ontologies. One of the benefits of using a system like SemNews is that it provides a mechanism for agents to populate various ontologies with live and updated information. While FOAF has become a very popular mechanism to describe a person's social network, not everyone on the web has a FOAF description. By linking the FOAF ontology to OntoSem's ontology we could populate additional information and learn new instances of foaf:person even though these were not published explicitly in foaf files but as plain text descriptions in news articles.

The SemNews environment also provides a convenient way for the users to query and browse the fact repository and triple store. Figure 21 shows a view that lists the named entities found in the processed news summaries. Using an ontology viewer the user can navigate through the news stories conceptually while viewing the instances that were found. The fact repository explorer provides a way to view the relations between different instances and see the news stories in which they were found. An advanced user may also query the triple store directly, using RDQL query language as shown in Figure 22. Additionally the system can also publish the RSS feed of the query results allowing users or agents to easily monitor new answers. This is a useful way of handling standing queries and finding news articles that satisfy a structured query.

Developing SemNews provided a perspective on some of the general problems of integrating a mature language processing system like OntoSem into a Semantic Web oriented application. While doing a complete and faithful translation of knowledge from OntoSem's native meaning representation language into OWL is not feasible, we found the problems to be manageable in practice for several reasons.

First, OntoSem's knowledge representation features that were most problematic for translation are not used with great frequency. For example, the default values, relaxable range constraints and procedural attachments were used relatively rarely in OntoSem's ontology. Thus shortcomings in the OWL version of OntoSem's ontology are limited and can be circumscribed.

Second, the goal is not just to support translation between OntoSem and a complete an faithful OWL version of OntoSem. It is unlikely that most Semantic Web content producers or consumers will use On-

toSem's ontology. Rather, we expect common consensus ontologies like FOAF, Dublin Core, and SOUPA to emerge and be widely used on the Semantic Web. The real goal is thus to mediate between OntoSem and a host of such consensus ontologies. We believe that these translations between OWL ontologies will of necessity be inexact and thus introduce some meaning loss or drift. So, the translation between OntoSem's native representation and the OWL form will not be the only lossy one in the chain.

Third, the SemNews application generates and exports facts, rather than concepts. The prospective applications coupling a language understanding agent and the Semantic Web that we have examined share this focus on importing and exporting instance level information. To some degree, this obviates many translation issues, since these mostly occur at the concept level. While we may not be able to exactly express OntoSem's complete concept of a book's author in the OWL version, we can translate the simple instance level assertion that a known individual is the author of a particular book and further translate this into the appropriate triple using the FOAF and Dublin Core RDF ontologies.

Finally, with a focus on importing and exporting instances and assertions of fact, we can require these to be generated using the native representation and reasoning system. Rather than exporting OntoSem's concept definitions and a handful of facts to OWL and then using an OWL reasoner to derive the additional facts which follow, we can require OntoSem to precompute all of the relevant facts. Similarly, when importing information from an OWL representation, the complete model can be generated and just the instances and assertions translated and imported.

Language understanding agents could not only empower Semantic Web applications but also create a space where humans and NLP tools would be able to make use of existing structured or semi structured information available. The following are a few of the example application scenarios.

Semantic Annotation and Metadata Generation

The growing popularity of folksonomies and social bookmarking tools such as del.icio.us have demonstrated that light-weight tagging systems are useful and practical. Metadata is also available in RSS and ATOM feeds, while some use the Dublin Core ontology. Some NLP and statistical tools such as SemTag[42] and the TAP[176] project aim to generate semantically annotated pages from already existing documents on the web. Using OntoSem in the SemNews framework we have been able to demonstrate the potential of large scale semantic annotation and automatic metadata generation. Figure 18 shows the graphical representation of the TMRs, which are also exported in OWL and stored in a triple store.

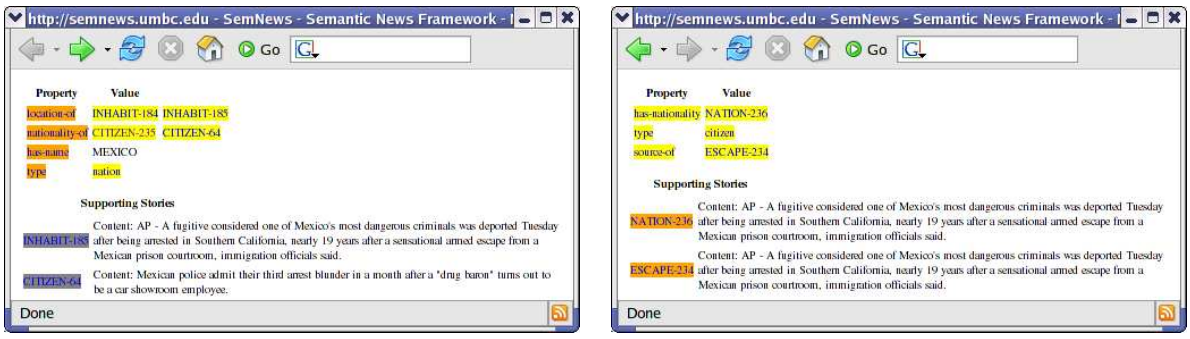


Figure 20: Fact repository explorer for the named entity 'Mexico'. Shows that the entity has a relation 'nationality-of' with CITIZEN-235. Fact repository explorer for the instance CITIZEN-235 shows that the citizen is an agent-of an ESCAPE-EVENT.

Gathering Instances

Ontologies for the Semantic Web define the concepts and properties that the agents could use. By making use of these ontologies along with instance data agents can perform useful reasoning tasks. For example, an ontology could describe that a country is a subclass of a geopolitical entity and that a geopolitical entity is a subclass of a physical entity. Automatically generating instance data from natural language text and populating the ontologies could be an important application of such technologies. For example, in SemNews you can not only view the different named entities as shown in Figure 21 but also explore the facts found in different documents about that named entity. As shown in 4., we could start browsing from an instance of the entity type 'NATION' and explore the various facts that were found in the text about that entity. Since OntoSem also handles referential ambiguities, it would be able to identify that an instance described in one document is the same as the instance described in another document.

Provenance and Trust

Provenance involves identifying source of information and tracking the history of where the information came from. Trust is a measure of the degree of confidence one has for a source of information. While these are somewhat hard to quantify and are a function of a number of different parameters, there can be significant indicators of trust and provenance already present in the text and could be extracted by the agent. News report typically describe some of the provenance information as well as other metadata that can effect trust such as temporal information. This type of information would be important in applications where agents need to make decisions based on the validity of certain information.

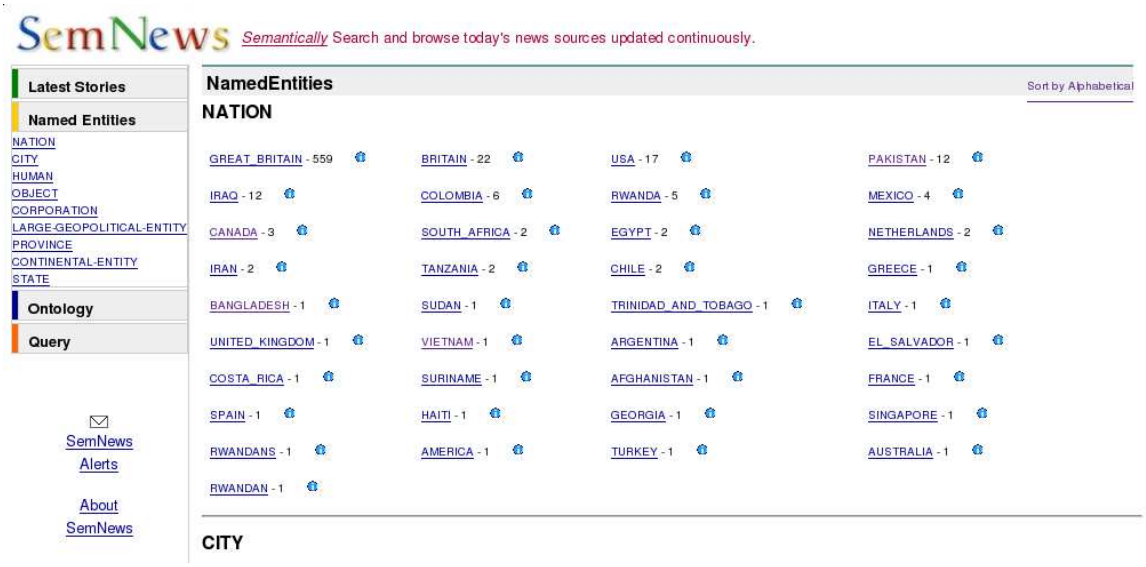
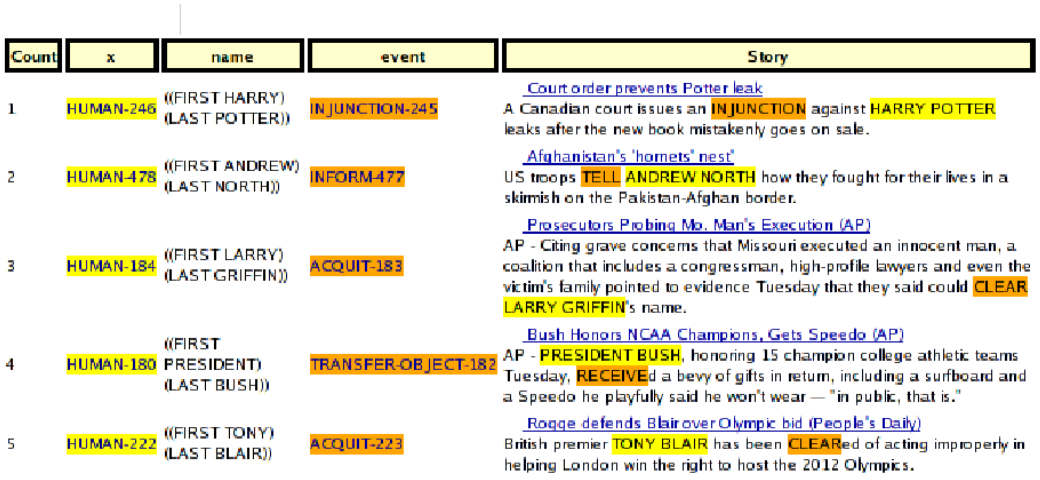


Figure 21: Various types of named entities can be identified and explored in SemNews.

Reasoning

While currently reasoning on the Semantic Web is enabled by using the ontologies and Semantic Web documents, there could be potentially vast knowledge present in natural language. It would be useful to build knowledge bases that could not only reason based on explicit information available in them, but also use information extracted form natural language text to augment their reasoning. One of the implications of using the information extracted from natural language text in reasoning applications is that agents on the Semantic Web would need to reason in presence of inconsistent or incomplete annotations as well. Reasoning could be supported from not just semantic web data and natural language text but also based on provenance. Developing measures for provenance and trust would also help in deciding the degree of confidence that the reasoning engine may have in the using certain assertions for reasoning.



Count	x	name	event	Story
1	HUMAN-246	((FIRST HARRY) (LAST POTTER))	INJUNCTION-245	Court order prevents Potter leak A Canadian court issues an INJUNCTION against HARRY POTTER leaks after the new book mistakenly goes on sale.
2	HUMAN-478	((FIRST ANDREW) (LAST NORTH))	INFORM-477	Afghanistan's 'hornets' nest' US troops TELL ANDREW NORTH how they fought for their lives in a skirmish on the Pakistan-Afghan border.
3	HUMAN-184	((FIRST LARRY) (LAST GRIFFIN))	ACQUIT-183	Prosecutors Probing Mo. Man's Execution (AP) AP - Citing grave concerns that Missouri executed an innocent man, a coalition that includes a congressman, high-profile lawyers and even the victim's family pointed to evidence Tuesday that they said could CLEAR LARRY GRIFFIN's name.
4	HUMAN-180	((FIRST PRESIDENT) (LAST BUSH))	TRANSFER-OBJECT-182	Bush Honors NCAA Champions, Gets Speedo (AP) AP - PRESIDENT BUSH, honoring 15 champion college athletic teams Tuesday, RECEIVEd a bevy of gifts in return, including a surfboard and a Speedo he playfully said he won't wear — "in public, that is."
5	HUMAN-222	((FIRST TONY) (LAST BLAIR))	ACQUIT-223	Rogge defends Blair over Olympic bid (People's Daily) British premier TONY BLAIR has been CLEARed of acting improperly in helping London win the right to host the 2012 Olympics.

Figure 22: This SemNews interface shows the results for query “Find all humans and what are they the beneficiary-of”

Chapter IV.

MINING SOCIAL MEDIA STRUCTURE

Communities are central to online social media systems and add to it's richness and utility. Detecting the community structure and membership is critical for many applications. The main approach used in analyzing communities has been through the use of the network structure. In this chapter, we present techniques for detecting communities that leaverage on the special properties and characteristics of social media datasets. Many social graphs follow a power law distribution. A few nodes attract most of the inlinks. We present a novel technique that can speed up the community detection process by utilizing this characteristic property of social media datasets. Many social applications support *folksonomies*, which provide users with the ability to tag content with free-form descriptive words. We describe an approach that combines the use of network structre and folksonomy or tag information to compute. Finally, in this Chapter, we also present an analysis of communities and user intentions in microblogging communities.

A. Approximate Community Detection in Social Media

While many factors can be used to identify communities, analyzing the network structure has been a key one. The problem is made difficult by the large size of the underlying graphs, making the algorithms typically used to identify their communities very expensive. We describe an approach to reducing the cost by estimating the community structure from only a small fraction of the graph using basic linear algebra manipulations and spectral methods. The technique exploits the fact that in most Web communities a small fraction of the members are highly linked while most (the "long tail") are sparsely connected to the network. It has the advantage of quickly and efficiently finding a reasonable approximation to the community structure of the

overall network. We also present an intuitive heuristic and show that it results in good performance at a much lower costs.

Communities are a central feature of social media system like blogs, social networking sites and online forums. Communities can form around and are shaped by many factors, including shared interests (knitting), common values or beliefs (network neutrality), or specific tasks or events (ICDE 2009). They add to the richness and utility of social media and are recognized as one of the distinguishing features of these systems. An important task in analyzing such networked information sources is to identify the significant communities that are formed.

A community in the real world is often reflected in the graph representation as a group of nodes that have more links within the set than outside it. In online applications, new entities are constantly created and discovered through Web crawls or creation of new links. Consider a meme tracker that automatically builds a list of popular posts in different categories like politics, technology, entertainment etc. Enabling quick classification of newly discovered resources into their respective communities can help identify the right category under which a new post should be placed. In this paper, we present a simple technique that lets us quickly approximate the community structure of entities in the long tail.

Our approach is based on an important assumption that large, scale-free networks are often very sparse. In addition they usually have a core-periphery network structure [17]. For many social networks and Web graphs, this has been found to be true [79]. Such networks consist of a small, but high degree set of core nodes and a very large number of sparsely connected peripheral nodes. In the head plus tail model, the peripheral nodes are found in the long tail. They can have a number of links into the core network, which is also justified by the preferential attachment model [5].

The insight behind our technique is that the community structure of the overall graph is very well represented in the core, and can be extracted from there alone. The community membership of the long tail can be approximated by first using the subgraph of the small core region to decide what communities exist, and then analyzing the connections from the long tail to the core. Figure 23 shows an example of a graph that consists of a collection of blogs. This is the adjacency matrix permuted so that the nodes with the highest degree are at the upper left corner, which forms the core for this network. The sub-matrix B corresponds to the links from peripheral nodes to the core, while C is the extremely sparse subgraph of connections among the peripheral nodes.

In the following sections we describe an approach that takes advantage of this sparsity to approximate the

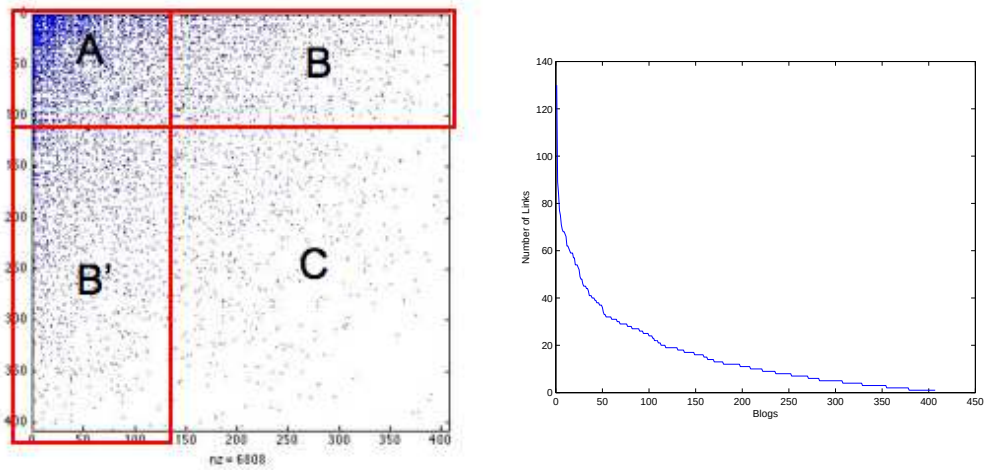


Figure 23: This adjacency matrix corresponds to a graph formed by 407 blogs monitored over a period of one year. The matrix is re-permuted so that high degree nodes are in the upper left corner. Sub-matrix A corresponds to the core of the network while B represents links from periphery (or long tail) nodes to nodes in the core. The graph on the right of the figure shows the distribution of degrees of the nodes in this network.

community structure of the complete matrix

$$\begin{pmatrix} A & B \\ B^T & C \end{pmatrix}$$

of size $\Re^{nxn}$ by using only a much smaller $\Re^{nxk}$ matrix,

$$\begin{pmatrix} A \\ B^T \end{pmatrix}$$

where $k << n$.

1. Related Work

A set of vertices can constitute a community if they are more closely related to one another than the rest of the network. Such vertices connect with a higher density within the group and are very sparsely connected to the rest of the network [156]. An intuitive measure for the quality of any clustering or community detection algorithm is the modularity function defined by Newman et al. [159]. The modularity function, Q, measures the fraction of all the edges, e_{ii} that connect within the community to the fraction of edges, a_i that are across communities. The measure Q is defined as

$$Q = \sum_i (e_{ii} - a_i^2) \tag{IV..1}$$

Determining the "best" community structure by finding the optimal modularity value has been shown to be NP-Hard [47] and is thus not a viable approach for even networks of relatively modest sizes. One of the earliest works on community detection was by Girvan and Newman [54]. They propose a divisive algorithm that works by removing edges with high betweenness centrality. By repeated recalculation of the edge betweenness and removal of edges the entire network is decomposed into its constituent communities. Other approaches include optimizing modularity via topological overlap [172], greedy heuristics [30] and using efficient update rules for merging clusters during the hierarchical clustering process [156].

Recently, spectral methods have been applied to community detection [27] and shown to have a relation to optimizing the modularity score [158]. Spectral clustering is a method that is based on the analysis of eigenvectors of a graph or more generally, any similarity matrix. It has been used to efficiently cluster data and partition graphs into communities. Shi and Malik [183] developed a normalized cut criteria to find balanced partition of image. Their proposed method optimizes the inter-cluster similarity as well as similarity within clusters. Though it was originally applied for image segmentation, spectral clustering has found several applications in graph mining and community detection. A comprehensive survey of spectral clustering is provided by von Luxburg [201].

Most spectral clustering techniques use either the unnormalized or normalized form of graph Laplacian. The graph Laplacian is a representation of the similarity matrix that has a number of important properties [65, 149]. The general format of a graph Laplacian is given by:

$$L = D - W \tag{IV..2}$$

where $W \in \Re^{n \times n}$ is the similarity matrix (or the adjacency matrix) and D is a diagonal matrix representing the degrees of nodes in the graph. The normalized version for the graph Laplacian is given by:

$$L_{norm} = D^{-\frac{1}{2}} W D^{-\frac{1}{2}} \tag{IV..3}$$

An important property of the graph Laplacian is that the smallest eigenvalue of L is 0 and the smallest non-zero eigenvalue corresponds to the *algebraic connectivity* of the graph [29]. The vector corresponding to the second smallest eigenvalue is also known as *Fiedler vector* [29]. The algebraic connectivity of the graph

is an indicator of how well connected the graph is. The original graph can be easily partitioned using only the sign of the values in the Fiedler vector.

A related method for community detection is using a graph kernel [181]. A kernel provides a simple mapping from the original, high dimensional feature space to a set of correlation scores or similarities between the data points (in the case of graphs nodes). There are many different formulations of a kernel [189].

Kernel methods have gained popularity in both statistical machine learning and pattern recognition literature [181]. Recently, they have also been applied to study graphs for applications like Web spam detection [209], semi-supervised learning [95] and classification to link analysis [114, 186]. We refer the reader to Ham et al. [73] for a review of kernel-based methods.

It is worth noting that the Girvan-Newman algorithm [54] and the kernel-based methods typically require $O(n^3)$ operations. The main bottleneck in the Girvan-Newman algorithm is the computation of all pairs shortest paths , while in kernel methods it is the calculation of an inverse. A reasonably fast and optimized code like NCut can approximately calculate a few eigenvectors using Lanczos method. METIS [101] is another example of a highly scalable graph clustering tool that works by partitioning the graph by iteratively coarsening and refining the communities found. The main difference between these techniques and our approach is that all these methods work by using the entire graph and hence have a high memory and performance requirement. In contrast our algorithm works by utilizing the inherent structure of social graphs and effectively approximates the communities of the entire network by using only a small portion of the graph.

2. Sampling Based Low Rank Approximations

We present an approach to quickly and approximately find the community structure in a network by selectively sampling a small subgraph from the whole network. The intuition is that scale-free networks are generally quite sparse and often consist of a core and periphery network. Compared to the rest of the network, the core is relatively small, but dense. The periphery nodes link to core nodes and there are very few links among the periphery network.

We start by first discussing an approach that was proposed by Drineas et al. [45]. Their approach is based on selectively sampling the columns of the original matrix to in proportion to their squared norm. In the blog graph matrix, the columns representing the nodes in the core will clearly be picked up disproportionately in any such sampling. Drineas' ideas is to find a rank k approximation to the original matrix by approximating the top k singular vectors. We describe the application of this technique to our problem following the descrip-

tion in Keuchel and Schnorr [83]. First, we permute the original adjacency matrix according to the degree (as shown in Figure 1). This makes use of the characteristic structure of social graphs. Permuting the adjacency matrix based on the degree focusses on the nodes in the core and samples of the columns of the adjacency matrix in a manner that conforms to the constraints imposed by [45].

Next, we compute the normalized Laplacian matrix (as described in Equation IV..3) associated with this graph. Note that since the original adjacency matrix, W, is sparse, L is also sparse. Also, it has the same (permuted) structure as W. Thus L can be partitioned into four sub-matrices as shown below:

$$\begin{pmatrix} A & B \\ B^T & C \end{pmatrix}$$

such that $A \in \Re^{k \times k}$, $B \in \Re^{k \times (n-k)}$ and $C \in \Re^{(n-k) \times (n-k)}$.

Recall that A represents the connectivity between nodes in the core, and B the connectivity of the nodes outside the core to those in the core. Now using singular valued decomposition (SVD) L can be factorized into it's singular values and corresponding basis vectors:

$$L = D^{-\frac{1}{2}} W D^{-\frac{1}{2}} = \sum_{i=1}^{n} \rho_i q_i p_i^T \tag{IV..4}$$

where ρ_i are the singular values, q_i and p_i are the left and right singular vector correspondingly. If Q_k is a matrix of left orthonormal singular vectors then the best k approximation of L is given by

$$L = Q_k * Q_k^T * L \tag{IV..5}$$

The approximate value for Q_k (left largest singular vectors) can be obtained by using the eigenvectors corresponding to the k largest eigenvalues of $S^T * S$ where the sub-matrix $S \in \Re^{nxs}$ is given by

$$S = \begin{pmatrix} A \\ B^T \end{pmatrix}$$

Let w_i be the eigenvectors corresponding the *k* largest eigenvalues of matrix $S^T * S$. Then the approximated Q_k of the L can be found by

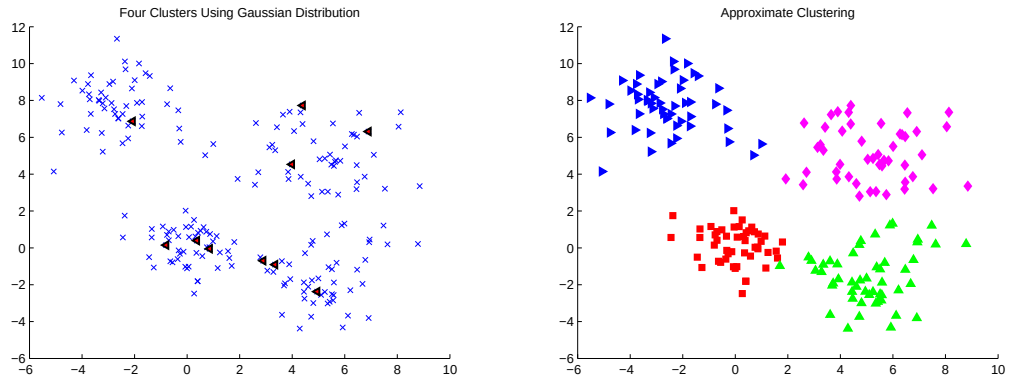


Figure 24: An illustrative example of sampling based approach. Four clusters are randomly generated using a Gaussian distribution. A few sample points suffice to approximately reconstruct the original clusters in the data.

$$qi = S * \frac{w_i}{||Swi||} = S * \frac{w_i}{\sqrt{\lambda i}} \; for \; i = 1.....k \tag{IV..6}$$

where λ_i denotes the eigenvalues of the matrix $S^T S$.

Since L is a positive semi-definite matrix, the singular values and singular vectors of L coincide with its eigenvalues and eigenvectors. This leads to Q_k approximating the k eigenvectors needed for community detection. A different interpretation of this approach is in terms of the Nyström method [51] is also presented in [83].

A simple illustration of the sampling based approximation is shown in Figure 24. Here four random clusters are generated using a simple Gaussian distribution. RBF Kernel is used to compute the pairwise distance between the points. A few points are randomly sampled from these clusters. Constructing the matrix S from the RBF kernel gives the distances between the sampled points (represented by A) and the distance of the remaining points to the sampled points (B). A few samples are sufficient to approximately reproduce the original clusters, as can be seen from this example.

Another possible way to approximately calculate communities (that has been sometimes used in literature) would be to cluster the singular vectors U obtained using the low-rank approximations of the original large, sparse matrix A. Given a matrix $L \in \Re^{m \times n}$, its Singular Valued Decoposition is given by

$$L = U * \sigma * V^T \tag{IV..7}$$

where $U \in \Re^{m \times m}$, $\sigma \in \Re^{n \times n}$ and $V^T \in \Re^{n \times n}$. In real applications, often the matrix L can be approximated

by its reduced rank SVD, with rank $k << n$ such that

$$L_k = U_k * \sigma_k * V_k^T \quad \text{(IV..8)}$$

In the above equation only k columns of the orthonormal basis vectors U and V^T are used along with k singular values in σ thereby making it more efficient to compute and store. This is also known as *truncated SVD* or *reduced rank approximation*.

Once the (approximate) SVD is known either using Drineas's approach or using the truncation method, the data is projected onto the k-dimensional subspace and clustered.

3. Heuristic Method

We propose another approach that uses the structure of the blogs graph directly. We use the head of the distribution (i.e. the highly connected nodes) to first find the communities in the graph. The intuition is that communities might form around popular nodes. So we can use any of the community detection algorithms to find the initial communities in a graph that is much smaller than the original one. This leaves the problem of finding the community of the blogs that are not a part of the head. One heuristic is to look at the number of links from a blog to each community as identified from the clustering of the nodes in the head, and declare it to be a member of the community that it most associates with by this measure. We present two such approaches in this paper. One uses Ncut to find communities in the head, and associate nodes in the tail to a community that it most frequently points to. This heuristic can significantly reduce the computation time, while providing a reasonable approximation to the community structure that would be found by running the same Ncut algorithm over the entire graph. Another takes the clustering approach of Drineas *et al.*, but projects onto the k-dimensional space formed by the SVD of the submatrix A.

4. Evaluation

We first present some empirical results using the intuition behind our technique – namely that a sampling of the entire graph can still lead to good community extraction. Consider Figure 25, which shows a dataset commonly used in community detection. Notice that as the fraction of the data sampled increases, the community structure becomes clearer.

For completeness, we compare our results with the normalized cut (Ncut) implementation by Shi et al.

Data	C	10%	30%	50%	Ncut
Political	4	0.20934	0.39805	0.50373	0.5237
Jazz	4	0.28070	0.41701	0.40382	0.4408
Football	11	0.16842	0.34018	0.45868	0.6020
NEC-Blogs	6	0.18285	0.28374	0.27910	0.2790
E-mail	15	0.29664	0.38588	0.44436	0.5498
PGP	80	0.45331	0.54481	0.57076	0.8605

Table IV..1: By sampling a small portion of the entire adjacency graph, a reasonable approximation of the community structure can be recovered at a very small cost. The mean modularity scores obtained over 30 iterations are reported here. The standard deviation for all runs was less than 0.1.

[183]. As described before, Ncut algorithm works by recursively partitioning the graph according to the second smallest eigenvector of the graph Laplacian.

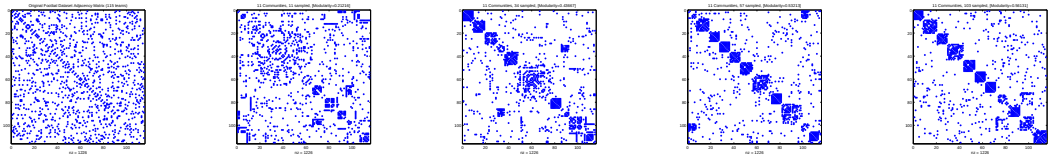


Figure 25: The football dataset consists of 115 teams. A pair of teams is linked if they have played a match against each other. The original matrix and communities found by sampling at 10%, 30%, 50% and at 80% are shown in the figure. Observe that as the size of the sampled matrix increases, so does the modularity score. Even at lower sampling, the overall modularity scores are still sufficient to identify some approximate communities.

The modularity scores give a measure of the goodness of clustering results. They indicate that the intra-cluster distances are less than the inter-cluster distance. However, we wish to verify that the communities that were found using the original, full matrix and the ones that were approximated are similar. One way to evaluate two different clustering results is to use the variation of information score [139]. Variation of information is a measure that describes the agreement of two community assignments. The variation of information between two cluster assignments is defined as

$$VI(C,C') = -\sum_{xy} p(x,y) log \frac{p(x,y)}{p(y)} - \sum_{xy} p(x,y) log \frac{p(x,y)}{p(x)} \quad \text{(IV..9)}$$

Where x,y are the labels assigned by the two clustering techniques C,C'. It attains a minimum value of zero when the two cluster assignments are identical and its maximum value is *log n*, where n is the number of items. Recently, this measure has also been used to evaluate the robustness of communities [99].

Table IV..2 summarizes these at different sampling rates. In this experiment we consider the resultant clusters using the entire matrix and NCut to be the ground truth. From the table, we can easily observe that as more columns are sampled, the resulting cluster assignments match those obtained by performing cuts on

Data	C	10%	30%	50%
Political	4	0.29984	0.18197	0.06526
Jazz	4	0.26322	0.16326	0.15737
Football	11	0.47134	0.24979	0.15586
NEC-Blogs	6	0.32347	0.30258	0.28075
E-mail	15	0.38634	0.33004	0.30912
PGP	80	0.38987	0.36921	0.3605

Table IV..2: Variation of Information measure for different datasets. The Ncut algorithm run on the entire matrix is considered to be the ground truth. The variation of information is minimized when the two cluster assignments are in exact agreement and can attain a value of $logn$ when there is no agreement.

the entire graph. Thus we can say that the approximations obtained are reasonable. Another point to observe is that the variation of information and modularity scores (from Table IV..1) show similar trends. It would be interesting to show the exact relation of variation of information score and modularity function. Intuitively, modularity is maximized when the cluster assignments group items within the same community closer to each other and vice-versa.

In order to evaluate the quality of approximation we use a blog graph consisting of six thousand nodes. Figure 28 shows the original sparse matrix permuted using the degree of the node to reveal the core-periphery structure of the graph also shows the communities detected using the heuristic method. Since there is no ground truth available, we use the modularity score, Q, as a measure of quality for the resulting communities found by each of the methods. We also compare the approximate methods with Ncut algorithm using variation of information score.

Figure 26 shows the performance of Ncut, low-rank SVD, approximation method and heuristic method for computing the communities. In the graph SVD stands for the rank k approximation using the truncated Singular Value Decomposition for the entire matrix. Approximation stands for Drineas' method. Heuristic corresponds to our approach of finding the communities using NCut on the head and approximating the communities for the long tail. NCut is Shi and Malik's Normalized Cut implementation run over the entire matrix. Since no ground truth is available for this dataset, the variation of information scores are reported by comparing the approximations to NCut on the entire matrix. The results indicate that both the approximation and heuristic method provide high modularity scores even at low sampling rates (10%-50%). Also, the time required to compute the communities is comparable or at times less than that of using Ncut. In addition the memory requirements are much less since only a small fraction of the entire graph is sampled.

Figure 27 shows the variation of information scores for the different methods. The variation of information scores measure the quality of the approximation compared to the communities detected via NCut.

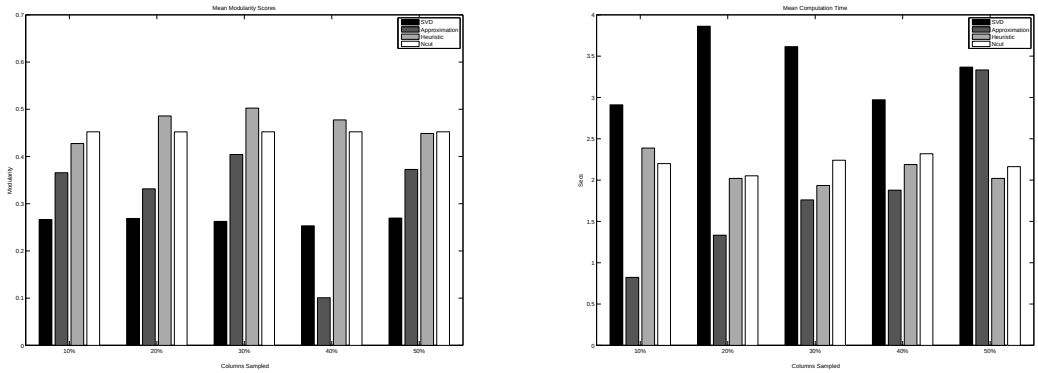


Figure 26: Modularity scores (left) and computation times (right) for different sampling rates (10% to 50%) over 50 runs. Bars are in the following order: SVD, Sampling Based Approximation, Heuristic, Ncut.

Interestingly, we find that the heuristic method performs very well. The advantage of the heuristic and sub-sampling based approximation is that it utilizes a much smaller matrix for computing the communities in the larger graph. Even at lower sampling rate, the modularity scores obtained are as good as those found by either performing Ncut on the entire graph or using the reduced rank approximation. Note however, that the variation of information using the heuristic is much lower than most of the other techniques. We suspect that this is because of the skewed distribution of the links. Nodes in the long tail are not as useful in distinctly identifying the communities and can add to noise when clustering using the entire matrix. Figure 28 shows the original sparse matrix and its corresponding community structure obtained from the heuristic method while using only 30% of the head nodes for the inital community detection.

Based on this initial analysis the results look promising. However one of the difficulties in evaluation that remains is the lack of accurate ground truth data. In addition several real world datasets are plagued with noise and spam [113]. This makes measuring the quality of the resulting clusters a challenge, and may require the use of robust clustering techniques.

In terms of running time the complexity of Ncut is $O(nk)$ where n is the number of nodes in the graph and k is the number of communities. Thus the heuristic is $O(rk)$ where r is the number nodes in the head. On the other hand, the complexity of SVD is $O(n^3)$ in general, however reducing a sparse matrix using k basis vectors typically requires less work. Finally the sub-sampling based approximation can be efficiently implemented in $O(r^3)$ using the Nyström Method [51] .

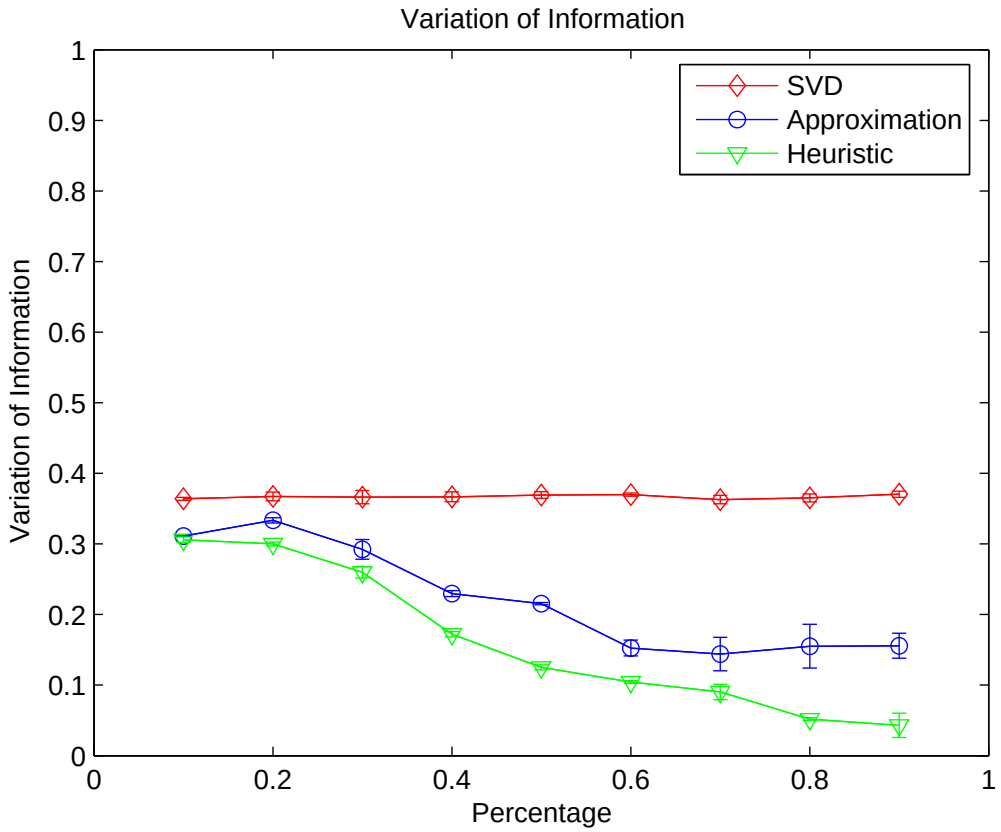


Figure 27: The Variation of information scores for the web graphs shown in Figure 28 using different approximation techniques. The mean and standard deviation values are shown over 10 runs.

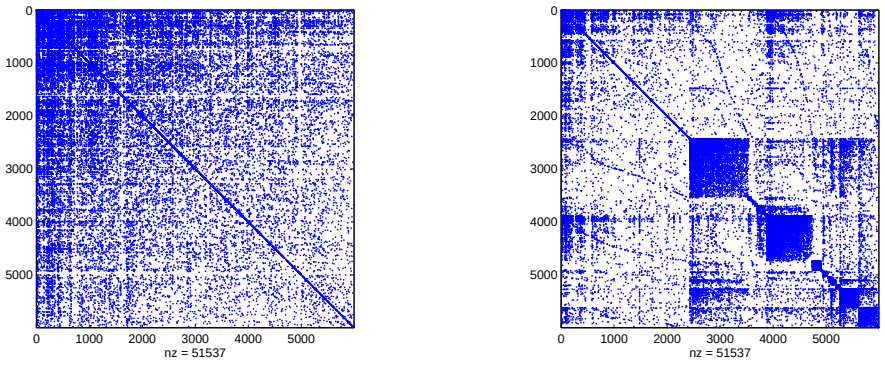


Figure 28: A Webgraph consisting of 6000 blogs (left) is sampled at 30% using the heuristic method. The resulting 20 communities identified are shown on the right. The modularity score was found to be about 0.51

5. Conclusions

It is challenging to identify and extract communities in very the large networks that characterize the online social media. One way to reduce the scale is to extract only a subgraph from the entire network. This makes it easier to analyze a small subset of nodes and relations at the cost of ignoring a very large portion of the nodes that are outside the core of the network. If we choose the subset carefully, the community structure we extract from it will be a good approximation for the community structure of the entire graph.

Given the power law distribution of most social media communities on the Web, we can focus on the small head of the distribution when computing the community structure. Once the community structure has been computed, members from the long tail can be added by a simple analysis of their links. The result is a significant reduction in the overall computational cost.

A key question about this approximation is whether important information about the community structure is lost by ignoring the contribution of the “long tail” of the distribution. We have applied our approach to a number of social media datasets with very encouraging results. The increased efficiency of the community detection algorithm is clear and our preliminary analyses of the quality of the detected communities shows it to be good when compared to using the entire graph. In ongoing work (results of which will be available for the final version of this paper if accepted) we are conducting a more formal evaluation of the approach.

B. Simultaneous Clustering of Graphs and Folksonomies

This section presents a simple technique for detecting communities by utilizing both the link structure and folksonomy (or tag) information. A simple way to describe our approach is by defining a community as a set of nodes in a graph that link more frequently within this set than outside it *and* they share similar tags. Our technique is based on the Normalized Cut (NCut) algorithm and can be easily and efficiently implemented. We validate our method by using a real network of blogs and tag information obtained from a social bookmarking site. We also verify our results on a citation network for which we have access to ground truth cluster information. Our method, Simultaneous Cut (SimCut), has the advantage that it can group related tags and cluster the nodes simultaneously.

Participants in social media systems like blogs and social networking applications tend to cluster around common topics of interest. An important task in analyzing such networked information sources is to identify the significant communities that are formed. Communities are one of the essential elements of social media and add to their richness and utility. A community in the real world is often reflected in the graph representation as a group of nodes that have more links within the set than outside it.

Many social media systems and Web 2.0 applications support free form tagging, also known as a *folksonomy*. A typical example of such a system is del.icio.us[1], where items are bookmarked with descriptive terms associated with the resource. Analysis of tagging systems has shown the utility of folksonomies in providing an intuitive way to organize, share and find information [75]. One approach to group related resources together is by utilizing the tag information. Two URLs belong to the same cluster if they are tagged or categorized under similar sets of tags. This approach was used by Java et al. [89] for clustering related blog feeds and to identify the popular feeds for a given topic.

Clustering based on tags or folksonomy exclusively misses the information available from the link structure of the Web graph. On the other hand, partitioning the graph based on links exclusively ignores tags and other user-generated meta data available in most social media systems. In this work, we address the problem of combining both the graph and folksonomy data to obtain significant communities in a social network or a blog graph. The intuition behind this technique is that a community is

> *a set of nodes in a graph that link more frequently within this set than outside it and they share similar tags.*

Figure 29 describes the above definition pictorially. The nodes in circles represent entities (URLs, blogs

[1] http://del.icio.us

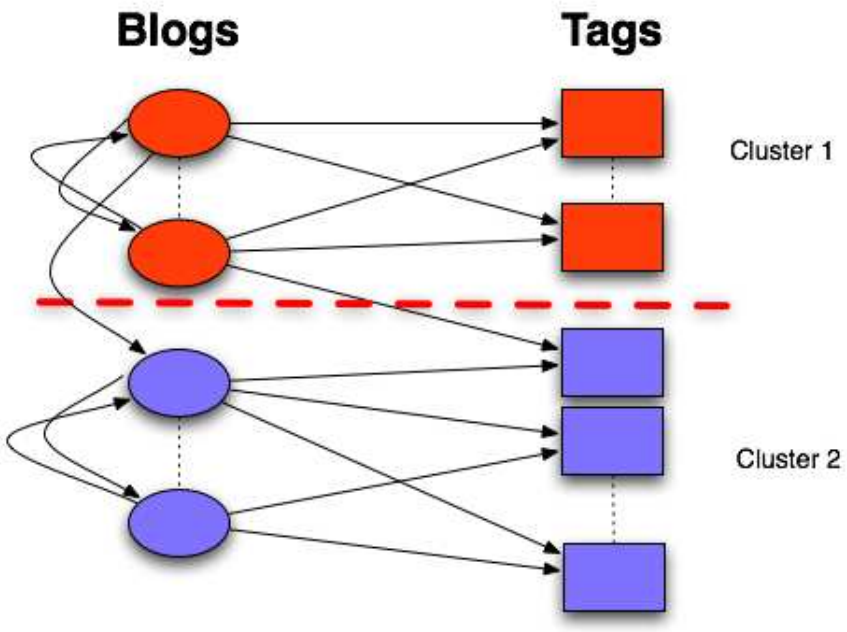


Figure 29: A community can be defined as a set of nodes in a graph that link more frequently within this set than outside it and the set shares similar tags.

or research papers). Such entities often link to each other via hyperlinks or citations. The square nodes represent the tag information or any user-generated content associated with a given resource. Several entities can share the same descriptive tags. Our extended definition of a community requires us to find a partition of the above graph such that it minimizes the number of edges cut in both the entity-entity and the entity-tag edge set. The Normalized Cut (NCut) algorithm [183] is an efficient technique to find these partitions. Our method, which is based on the NCut algorithm, can be efficiently implemented and provides a good clustering of the graph into its constituent communities.

1. Related Work

However, this technique for finding communities relies entirely on the link structure. In social media, there are a number of additional sources of meta-data information and annotation that can be obtained. Folksonomies or tags are one form of user-generated meta-data. There can possibly exist many more features that can be additionally used to identify communities. A few examples of these are sentiments and link polarity [96], related Wikipedia entries [194], links to main stream media sites, comments in blog posts, tags used by the blogger (as opposed to the tags used by readers or in social bookmarking sites). All these features provide additional cues and can be potentially useful in community detection algorithms. However, it is not always

clear how to integrate these into an unsupervised learning method.

A closely related clustering technique is co-clustering [41]. Co-clustering works by mapping an $m \times n$ term document matrix, A into a bipartite graph. The adjacency matrix of the bipartite graph is represented as

$$M = \begin{pmatrix} 0 & C \\ C^T & 0 \end{pmatrix}$$

where $C_{ij} = 1$ if the word j occurs in document i. It was shown [41] that the optimal clustering can be found by partitioning the graph represented by M. However, note that in this technique the links between the document set are never used. In the following section, we will describe the relation of our methods with the co-clustering.

2. Clustering of Graph and Tags

Our approach for simultaneously clustering graphs and tags was inspired by the classification constrained dimensionality reduction method proposed by Costa and Hero [34] and the co-clustering algorithm proposed by [41]. The constrained dimensionality reduction technique tries to incorporate the class label information to represent a high dimensional data into a lower dimensional space. For example, if the goal was to classify the collection of documents, using the approach presented by Costa and Hero, the known class labels (from the training data) are incorporated into the dimensionality reduction step. The algorithm optimizes a cost function such that the class estimates for the training data is close to the final cluster center and it also satisfies the normalized cut criteria. Their approach belongs to a general class of semi-supervised learning methods.

Following the notations used by Costa and Hero [34], let $W \in \Re^{n \times n}$ represent the adjacency matrix for a set of n nodes. Let $C \in \Re^{n \times k}$ be a matrix that represents if a node is associated with one of the k tag and $\beta > 0$ be a scaling parameter that regulates which of the two information link structure or tags is given more importance. Then the partitioning of the nodes into communities such that the membership is determined based on both the link structure and tags can be found by the eigenvectors associated with the matix

$$W^{'} = \begin{pmatrix} I & C \\ C^T & \beta W \end{pmatrix}$$

The matrix W' combines information from both the graph and the folksonomy. The first k columns correspond to the entity-tag edges in Figure 29 while the last n columns represent the entity-entity links.

Table IV..3: Table summarizing the statistics for the data used in this experiment. The first dataset is a paper citation network while the other is a blog graph network. Both datasets are comparable in size.

Citeseer Data		
1	Number of Papers	3312
2	Number of Words	3703

Blog Data		
1	Number of Documents	3286
2	Number of Tags	3047
3	Number of Homepages	3111
4	Number of stemmed words	10191

Finding a partition in the above graph that minimizes the number of edges that are cut, will result in clusters that have more links within the set than outside it and at the same time share similar sets of tags. This satisfies our extended definition of a community. Also note the relation to co-clustering in the above matrix. If the parameter β is set to 0, it would lead to the bipartite graph model used by Dhilon [41]. In our experiments that follow, we set $\beta = 1$ indicating an equal importance to tag information and graph structure.

A related technique is the constrained spectral clustering approach discussed in Xu et al. [206]. Their work utilizes the pairwise constraint information that describe if two nodes *must-link* or *cannot-link* [202]. In some cases this information can be available from domain knowledge or directly derived from the data.

3. Dataset Description

The following section presents the experimental results on two datasets. One is a network of academic paper citations and the associated text with these publications. This dataset contains six clusters for which ground truth label information is available. The other dataset is a blog graph network and the corresponding folksonomy extracted from a social bookmarking site.

For our experiments, we have used two datasets, summarized in Table IV..3. The first dataset is a citation network of academic publications derived from Citeseer[2] [14]. It consists of 3286 papers from six different categories: Agents, Artificial Intelligence (AI), Databases (DB), Human Computer Interaction (HCI), Information Retrieval (IR) and Machine Learning (ML). The category information was provided in the dataset along with a binary document-term matrix indicating the presence or absence of a term in a given publication. Since, this dataset has the ground truth for classification, it makes it ideal for our experiments. Since we do not have any folksonomy information associated with the publications, we use the words as a substitute for tag information. Since only a binary term vector for each document is provided in this collection, we use an Radial Bias Function (RBF) kernel, $K_{ij} = exp(-||x_i - x_j||^2/2\sigma^{-2})$ to compute the document similarities.

The second dataset is a subset of the Weblogging Ecosystems (WWE) workshop dataset. The original

[2]http://citeseer.ist.psu.edu/

dataset consists of about 10M posts from 1M weblogs over a 20 week period. From the original dataset, we extracted a subgraph corresponding to the top five thousand high PageRank nodes. Next, for each of these blogs we fetched the tags associated with its URL in del.icio.us, a social bookmarking tool. We found 3286 blogs that had some tags associated with them in this system. We chose to use the folksonomy from del.icio.us since it is currently the most popular social bookmarking tool. As opposed to self-identified tags specified by the blogger in blog search engines like Technorati[3], del.icio.us ranks the most popular tags associated with a given URL is aggregated over several users. User-generated labels or tag information provide descriptive meta-data that are helpful in determining the topic or theme of a resource and hence can be helpful in community detection. In general, we can extend our method to use any additional meta-data such as opinions, machine learned categories, etc. Although both the datasets contain directed edges, for our analysis we have convert the graph into an undirected network. This was primarily done due to ease of computation of Normalized Cuts over undirected representation of the graph. As future work, we plan to use directed, weighted normalized cut algorithm [140] that may be more applicable for Web graphs and citation networks.

Finally for the 3286 blogs, the corresponding homepages (or cached versions when available in Google) were downloaded. There were in all 3111 homepages that were retrievable. Since this dataset was originally obtained from a crawl performed in 2005, some of the homepages were non-existent. Using the set of available homepages, a hundred topics were learned using the Latent Dirichilet (LDA) model [15]. This was done primarily as a means for dimensionality reduction. Previously, LDA has been used in clustering blogs and has been shown to be an effective tool in summarizing the key topics [151].

4. Evaluation

First we present some empirical results using the blog dataset. The NCut algorithm partitions the graph to determine a set of communities by using only the link information. Once the communities are determined we would like to identify the tags associated with each of these communities. We use a simple approach of identifying the most frequently occurring tags in a given community. Table IV..4 presents the top five tags associated with 10 communities (out of 35) as identified using NCut.

One advantage of using SimCut over NCut algorithm is that it can be effectively used to cluster both the blogs and the tags simultaneously. Table IV..5 presents the top five tags associated with 10 communities

[3] http://technorati.com

Table IV..4: Top five tags associated with 10 communities found using NCut. For each community the most frequently used tags are shown in this table.

1	blog, blogs,technology, news, web
2	blog , poet, tags, browser, sustainability
3	blog, blogs, news, conspiracy, patterns
4	blog, blogs, kids, china, parenting
5	blog, crafts, craft, blogs, crafty
6	tutorials, graphics, webdesign, design, blogs
7	blog, programming, news, forum, .net
8	blog, cinema, french, literature, religion
9	blog, blogs, music, culture, art
10	blog, knitting, blogs, knitblogs, knitblog

Table IV..5: Top five tags associated with 10 communities found using SimCut.

1	food, cooking, recipes, foodblog, foodblogs
2	technology, business, web2.0, marketing, advertising
3	israel, jewish, judaism
4	christian, religion, philosophy, christianity, church
5	knitting, knitblogs, knitblog, knit
6	law, economics, legal, academic, libertarian
7	blogs, daily, culture, humor, funny
8	politics, media, liberal, political, progressive
9	design, web, webdesign, inspiration, css
10	tech, geek, gadgets, games, computer

(out of 35) as identified using SimCut. Empirically, the tags associated with the communities form coherent clusters and can be easily associated with the general theme of the community.

Next we look at some of the statistics of communities extracted using the two methods discussed. First, we present results on the citeseer citation network. Given that the hand-labeled ground truth information is available, this dataset has the advantage that the results can be compared to the actual communities present in the graph. Tables IV..6 and IV..7 show the confusion matrix for the two clustering methods. The results indicate that while the clusters are easily identifiable using SimCut approach, the NCut approach fails to find the right clusters. In general NCut finds very large partitions in the graph that are determined by using the link information alone. The overall accuracy obtained using SimCut is around 62%.

Figure 30 shows the average cluster similarity for 6 clusters extracted using NCut and SimCut algorithms on the citeseer dataset and the distribution of the community sizes. The similarity scores were obtained by averaging the inter-document scores obtained from the RBF kernel over the term document matrix. The average cluster similarity is computed as follows:

$$\frac{\sum_{d_i \in C, d_j \in C'} K(d_i, d_j)}{p} \tag{IV..10}$$

Table IV..6: Confusion matrix for NCut on Citeseer data giving an overall accuracy of 36%

	NCut					
	IR	**HCI**	**DB**	**AI**	**ML**	**Agents**
1	**461**	50	81	29	182	19
2	0	**2**	9	2	0	0
3	122	154	**186**	93	199	82
4	45	1	174	**22**	2	8
5	2	0	66	1	**44**	0
6	38	301	251	104	163	**487**

Table IV..7: Confusion matrix for SimCut on Citeseer data giving an overall accuracy of 61.7%

	SimCut					
	AI	**IR**	**HCI**	**ML**	**Agents**	**DB**
1	**70**	44	22	78	95	166
2	4	**366**	19	24	4	30
3	23	49	**359**	35	29	16
4	77	104	15	**372**	1	25
5	62	32	66	60	**430**	18
6	13	73	27	21	24	**446**

where $K(d_i, d_j)$ represents the score from RBF kernel and p corresponds to the number of such comparisons. Figure 31 depicts the clusters obtained by the two methods and reflects the true size of the communities found. Notice that NCut results provide a few very small communities while most communities are large and have a relatively low average document similarity score. Finally Figure 32 shows the clusters and sparsity plots obtained by reordering the original adjacency matrix using true cluster labels, NCut Communities and SimCut communities.

Figure 33 shows the average cluster similarity for 35 clusters extracted using NCut and SimCut algorithms for the blog dataset. One difficulty in evaluation for this data set is the lack of availability of any "*ground truth*" information. In order to circumvent this problem we have used the text from the blog homepages as a substitute. However, one thing to note is that this can be subject to a lot of noise that is typically contributed by various elements present on the homepage: navigation menus, advertising content, blogging platform specific templates etc [91]. Using the LDA algorithm, text from the homepages was mapped to topics vectors. The scores represented in the figure reflect the average similarities between the topic vectors for each blog.

From the distribution of community sizes we can find that the NCut algorithm results in partitions that lead to a few large communities and several very small communities. This can be explained by the fact that the NCut algorithm only uses the link information and it does not have the additional meta-data (via tag information that is available to the SimCut algorithm). In comparison the SimCut algorithm finds several communities of moderate sizes. NCut yields several very small or tightly knit communities of high similarity and a few large communities of very low similarity.

One benefit of using the SimCut algorithm is that even if a few links are missed due to crawling or parsing issues, it can still find the appropriate membership information since it relies on the additional feature of tags to compare the two documents. Finally Figure 35 shows the sparsity plots for the communities found using

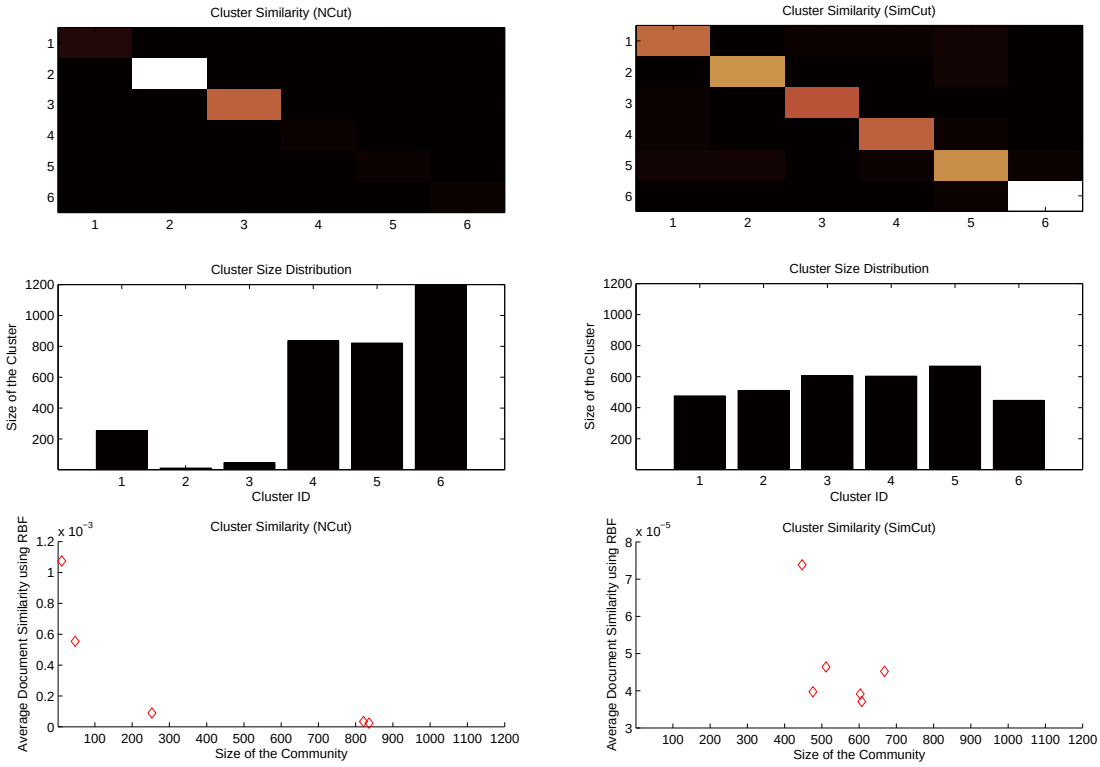


Figure 30: The above graphs show the average cluster similarity and size distributions of the communities found using NCut and SimCut. The NCut algorithm obtains a few very large communities and a large number of very small ones. In contrast the sizes of the communities found using SimCut is more balanced.

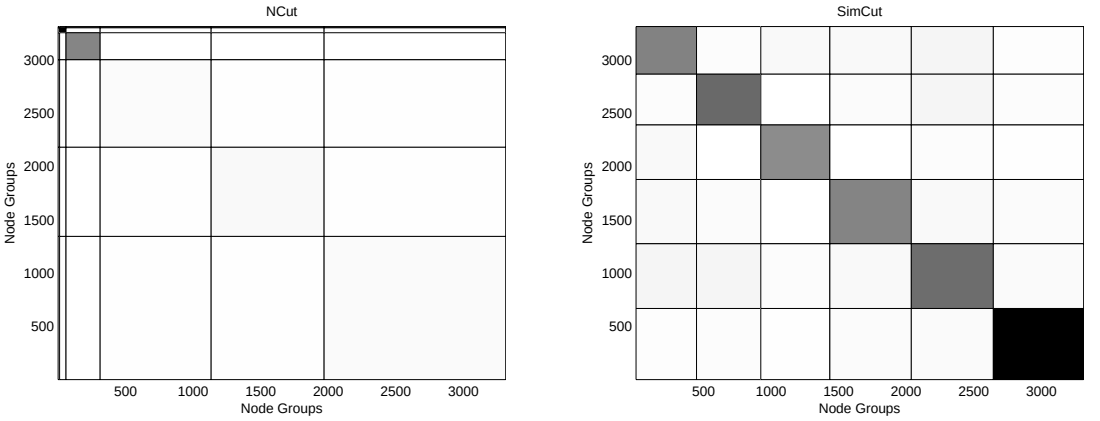


Figure 31: 6 Clusters obtained using NCut and SimCut algorithm on the citeseer dataset. Each square in the diagonal corresponds to the communities. The shade of the squares represents the average inter/intra cluster similarity scores.

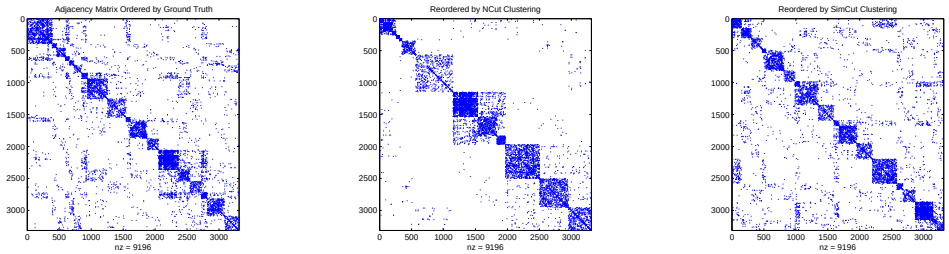


Figure 32: The above sparsity plots show the adjacency matrix ordered by a) the true cluster labels b) Communities found by NCut approach and c) Communities found by SimCut.

the two techniques. A point to note here is that although the SimCut criteria does not directly optimize for the modularity score, it does not degrade it significantly either. For example in the clustering results shown in this figure, for 35 communities, the modularity scores [158] determined using NCut is 0.4939 and the corresponding value using SimCut is 0.486. We use 35 communities since it resulted in the best modularity scores over a number of empirical runs.

Given the difficulty and high cost (two to three minutes per blog) of providing human annotation and judgement for the clustering results, one way to verify the performance of the two algorithms is to use the topic vectors generated by LDA. We construct a similarity matrix, $K \in \Re^{n \times n}$, where n is the number of documents (blog homepages). We use the NCut algorithm to identify the clusters in this document similarity matrix. If there was no link or tag information available, this would be the 'best' that we can approximate the ground truth without manually annotating each blog. Table IV..9 compares the effect of adding tag information and varying the number of clusters. In order to compare the two clustering techniques, NCut and SimCut we use the clusters found using the topic vectors as the "ground truth". Normalized Mutual Information is used to obtain the distance measure between the two clustering results. Mutual information between two clustering results C, C' is defined as

$$MI(C, C') = \sum_{c_i \in C, c'_j \in C'} p(c_i, c'_j) \log_2 \frac{p(c_i, c'_j)}{p(c_i).p(c'_j)} \tag{IV..11}$$

where $p(c_i)$, $p(c'_j)$ are the probabilities that an arbitrary document belongs to cluster c_i and c'_j respectively. The Normalized Mutual Information score is a value between 0 and 1 that represents how close two clustering results are.

From the results shown in Figures IV..8 and IV..9, we can find that the normalized mutual information increases as more tags. For example, the score is highest at around 35 communities determined using 500 tags, in the case of the blog dataset. However, adding even more tag information does not help. The mutual

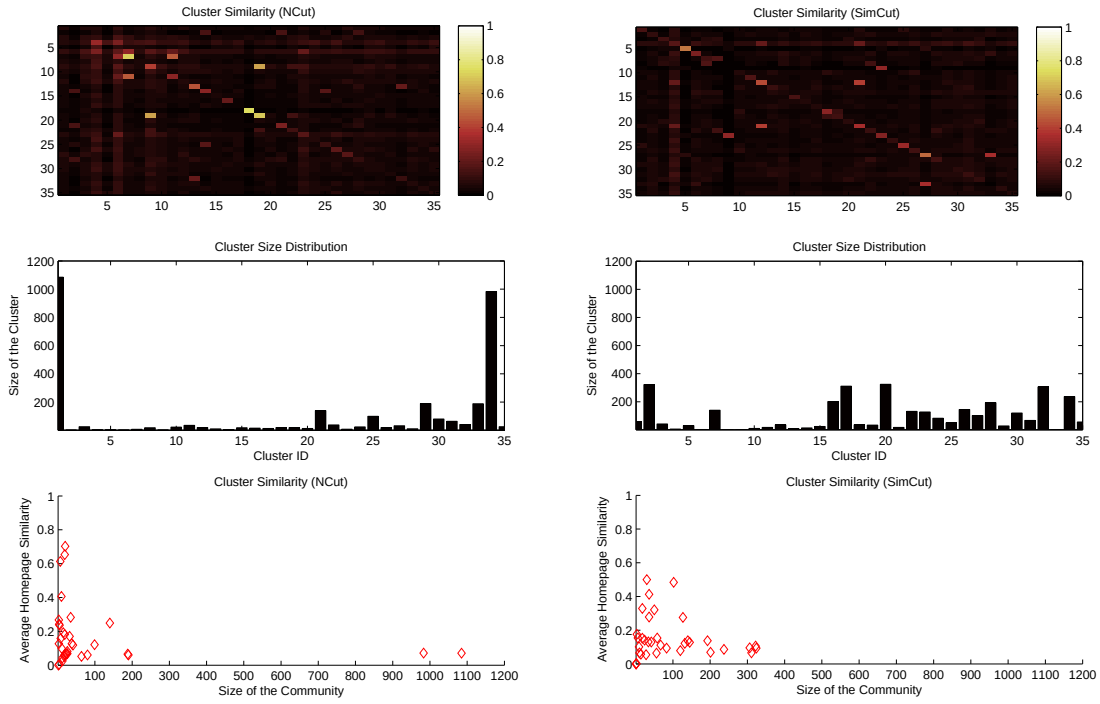


Figure 33: The above graphs show the average cluster similarity and size distributions of the communities found using NCut and SimCut. The NCut algorithm obtains a few very large communities and a large number of very small ones. In contrast the sizes of the communities found using SimCut is more balanced.

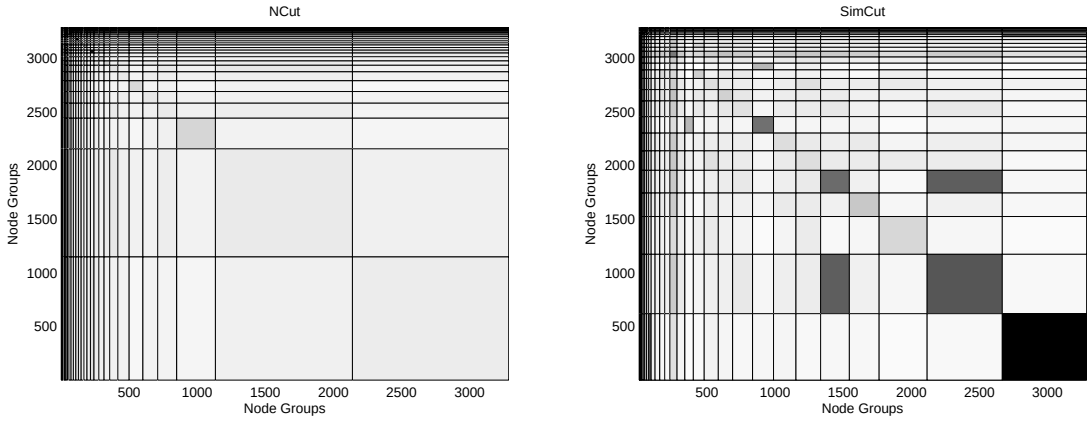


Figure 34: 35 Clusters obtained using NCut and SimCut algorithm. Each square in the diagonal corresponds to the communities. The shade of the squares represents the average inter/intra cluster similarity scores.

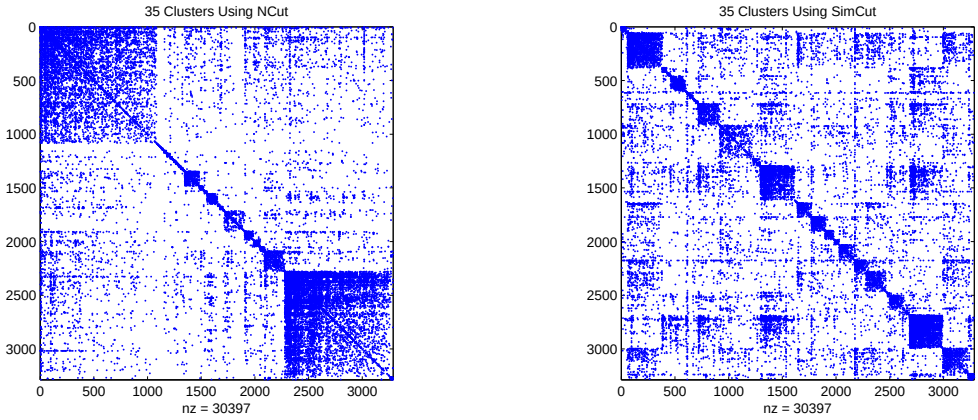


Figure 35: The above sparsity graphs show the communities found using the two clustering approaches. The original graph of 3286 nodes was first partitioned into 35 communities using NCut. Next, by adding the top 500 tags from del.icio.us, a social book marking site, the SimCut algorithm constrains the partitions such that a communities also share similar labels or tags, thus resulting in better clustering.

Table IV..8: Table Summarizing the Normalized Mutual Information Scores for citeseer dataset as more words are used in determining the clusters. Values reported here are averaged over 10 runs.

		SimCut (Number of Words Used)			
Clusters	NCut	50	200	500	1000
2	0.16293	0.1822	0.31934	0.35692	0.35071
3	0.16283	0.18196	0.31921	**0.35694**	0.35021
4	0.16443	0.18106	0.31949	0.35670	0.35042
5	0.16443	0.18161	0.31946	0.35665	0.35030
6	0.16126	0.17801	0.31942	**0.35682**	0.35019

information is higher than the clusters found using the link graph alone.

5. Discussion

The modularity score is a well accepted measure of community structure commonly used in the current literature on this subject. One key question that has been a topic of much debate recently is that of the statistical significance of modularity scores.

Statistical significance tests measure if a particular result is an effect of random chance. In some sense, the modularity score inherently compares the community label assignments to that of random graphs with similar configurations. Modularity score is a measure that compares the fraction of intra-community edges with that of expected number of edges if the label assignments were random. Thus a higher modularity score indicates that the graph exhibits a stronger community structure, which is less likely to be expressed by a random graph with similar number of nodes and edges.

Some authors [69] have recently suggested the use of z-score to measure if the modularity scores obtained

Table IV..9: Table Summarizing the Normalized Mutual Information Scores for blog dataset as more tag information is used in determining the clusters. Values reported here are averaged over 10 runs.

		SimCut (Number of Tags Used)			
Clusters	NCut	50	200	500	1000
25	0.20691	0.22720	0.27000	0.27970	0.25878
30	0.20693	0.22615	0.27109	0.27978	0.25928
35	0.20901	0.22521	0.26998	**0.2803**	0.25791
40	0.20895	0.22584	0.27208	**0.2800**	0.25840
45	0.20861	0.22503	0.27090	0.27938	0.26004
50	0.20986	0.22767	0.27139	0.27954	0.25633

for a network is sufficiently different and significantly higher from that of random graphs. The approach requires generating a number of random graphs and computing the mean μ and standard deviation σ of the modularity scores of these graphs. The z-score is then defined as

$$z = \frac{Q - \mu}{\sigma} \tag{IV..12}$$

Where Q is the modularity score of a given network and z measures the number of standard deviations Q is from the modularity scores of random graphs of similar configurations. A high value of z score indicates statistically significant modularity scores.

It has been shown by Karrer et al. [100] that the z-score measure lacks the reliability to be used as a trusted metric of significance test. They show that certain networks that exhibit strong community structures can often have low z-scores, thus making this metric unreliable. According to Karrer et al., the significance of communities found by an algorithm can be tested in terms of robustness of the results in presence of some random perturbations. Intuitively, a good community detection technique should not change results if a few random edges are modified in the network. Thus to measure the robustness of a community detection algorithm, a given network is randomly rewired, one edge at a time, so the new network has similar number of nodes, edges as the original network. By comparing the community structures found in the rewired network with that of the original network, a measure of robustness can be determined. Similar to our approach, the variation of information measure is used to compare label assignments of the communities found in the original network with those in the rewired network.

Another promising approach to measure the statistical significance of the results is to consider this as a sampling problem. Given a community assignment, the goal is to compare the distribution of community sizes and other properties like the power-law degree distributions with those of random graphs. Comparing two sample distributions drawn from the original network and the approximated network using the Kolmogorov-

Smirnov or D-static measure similar to the approach presented in Leskovec et al. [122] could be a simple non-parametric significance measure.

In our work, the goal is to measure the statistical significance of the quality of approximation of the community structure compared to the ground truth or the baseline method. In SimCut algorithm the goal is to measure the significance of the communities detected by adding the extra tag information compared to the communities found without using any tag information. The applicability of t-test and other non-parametric significance tests in such scenarios is currently an open research question.

6. Conclusions

Many social media sites allow users to tag resources. In this work, we have shown how incorporating folksonomy information in calculating communities can yield better results. The SimCut algorithm presented in this chapter is based on the Normalized Cut algorithm and can be easily extended to include additional user-generated meta-data (ratings, comments, tags in blog posts, etc). A key advantage of our approach is that it clusters both the tags and graph simultaneously. One challenge in community detection algorithms is that of labeling. Providing the right label that identifies the community is beneficial in visualization and graph analysis. We are currently investigating how our technique could be used to provide intuitive labels for communities. Finally, we are focussing our study on extending SimCut to weighted, directed networks.

C. Microblogging Communities and Usage

Microblogging is a new form of communication in which users describe their current status in short posts distributed by instant messages, mobile phones, email or the Web. We present our observations of the microblogging phenomena by studying the topological and geographical properties of the social network in Twitter, one of the most popular microblogging systems. We find that people use microblogging primarily to talk about their daily activities and to seek or share information. We present a taxonomy characterizing the the underlying intentions users have in making microblogging posts. By aggregating the apparent intentions of users in implicit communities extracted from the data, we show that users with similar intentions connect with each other.

Microblogging is a variation on blogging in which users write short posts to a special blog that are subsequently distributed to their friends and other observers via text messaging, instant messaging systems, and email. Microblogging systems first appeared in mid 2006 with the launch of Twitter[4] and have multiplied to include many other systems, including Jaiku[5], Pownce[6], and others. These systems are generally seen as part of the "Web 2.0" wave of applications [64] and are still evolving.

Microblogging systems provide a light-weight, easy form of communication that enables users to broadcast and share information about their current activities, thoughts, opinions and status. One of the popular microblogging platforms is Twitter [169]. According to ComScore, within eight months of its launch, Twitter had about 94,000 users as of April, 2007 [32]. Figure 36 shows a snapshot of the first author's Twitter homepage. Updates or posts are made by succinctly describing one's current status through a short message (known in Twitter as a *tweet*) that is limited to 140 characters. Topics range from daily life to current events, news stories, and other interests. Instant messaging (IM) tools including Gtalk, Yahoo and MSN have features that allow users to share their current status with friends on their buddy lists. Microblogging tools facilitate easily sharing status messages either publicly or within a social network.

Microblogging differs from conventional blogging in both how and why people use it. Compared to regular blogging, microblogging fulfills a need for a faster and more immediate mode of communication. In constraining posts to be short enough to be carried by a single SMS (Short Message Service) message, microblogging systems also lower a user's investment of the time and thought required to generate the content. This also makes it feasible to generate the content on the limited keypads of mobile phones. The reduced

[4] http://www.twitter.com
[5] http://www.jaiku.com
[6] http://www.pownce.com

Figure 36: A Twitter member's recent microblogging posts can be viewed on the web along with a portion of the member's social network. This example shows the page for the first author with posts talking about his daily activities, thoughts and experiences.

posting burden encourages more frequent posting – a prolific blogger may update her blog every few days whereas a microblogger might post every few hours. The lowered barrier also supports new communication modes, including what one social media researcher [174] calls ambient intimacy.

> *Ambient intimacy is about being able to keep in touch with people with a level of regularity and intimacy that you wouldn't usually have access to, because time and space conspire to make it impossible.*

While the content of such posts ("I'm having oatmeal with raisins for breakfast") might seem trivial and unimportant, they are valued by friends and family members.

With the recent popularity of microblogging systems like Twitter, it is important to better understand *why* and *how* people use these tools. Understanding this will help us evolve the microblogging idea and improve both microblogging client and infrastructure software. We tackle this problem by studying the microblogging

phenomena and analyzing different types of user intentions in such systems.

Much of research in user intention detection has focused on understanding the intent of a search queries. According to Broder [19], the three main categories of search queries are navigational, informational and transactional. Navigational queries are those that are performed with the intention of reaching a particular site. Informational queries are used to find resources on the Web such as articles, explanations etc. Transactional queries serve to locate shopping or download sites where further interactions would take place. According to this survey, most queries on the Web are informational or transactional in nature. Kang and Kim [97] present similar results in a study using a TREC data collection.

Understanding the intention for a search query is very different from user intention for content creation. In a survey of bloggers, Nardi et al. [153] describe different motivations for "why we blog". Their findings indicate that blogs are used as a tool to share daily experiences, opinions and commentary. Based on their interviews, they also describe how bloggers form communities online that may support different social groups in real world. Lento et al. [121] examined the importance of social relationship in determining if users would remain active users of the Wallop blogging system. A user's retention and interest in blogging was predicted by the comments received and continued relationship with other active members of the community. Users who are invited by people with whom they share pre-exiting social relationships tend to stay longer and active in the network. Moreover, certain communities were found to have a greater retention rate due to existence of such relationships. Mutual awareness in a social network has been found effective in discovering communities [128].

In computational linguistics, researchers have studied the problem of recognizing the communicative intentions that underlie utterances in dialog systems and spoken language interfaces. The foundations of this work go back to Austin [7], Stawson [193] and Grice [62]. Grosz [66] and Allen [6] carried out classic studies in analyzing the dialogues between people and between people and computers in cooperative task oriented environments. More recently, Matsubara [135] has applied intention recognition to improve the performance of automobile-based spoken dialog system. While their work focuses on the analysis of ongoing dialogs between two agents in a fairly well defined domain, studying user intention in Web-based systems requires looking at both the content and link structure.

In this section, we describe how users have adopted the Twitter microblogging platform. Microblogging is relatively nascent, and to the best of our knowledge, no large scale studies have been done on this form of communication and information sharing. We study the topological and geographical structure of Twitter's

social network and attempt to understand the user intentions and community structure in microblogging. Our analysis identifies four categories of microblogging intention: daily chatter, conversations, sharing information and reporting news. Furthermore, users play different roles of information source, friends or information seeker in different communities. We would like to discover what makes this environment so different and what needs are satisfied by such tools. In answering some of these questions, we also present a number of interesting statistical properties of user behavior and contrast them with blogging and other social network.

1. Dataset Description

Twitter is currently one of the most popular microblogging platforms. Users interact with this system by either using a Web interface, instant messaging agent or sending SMS messages. Members may choose to make their updates public or available only to friends. If user's profile is made public, her updates appear in a "public timeline" of recent updates and distributed to other users designated as friends or followers. The dataset used in this study was created by monitoring this public timeline for a period of two months, from April 01, 2007 to May 30, 2007. A set of recent updates were fetched once every 30 seconds. There are a total of 1,348,543 posts from 76,177 distinct users in this collection.

When we collected our data, Twitter's social network included two types of directed links between people: friend and follower. A Twitter user can "follow" another user, which results in their receiving notifications of public posts as they are made. Designating a twitter user as a friend also results in receiving post notifications, but indicates a closer relationship. The directed nature of both relations means that they can be one-way or reciprocated. The original motivation for having two relationships was privacy – a microblogger could specify the some posts were to be visible only to her friends and not to her (mere) followers. After the data was collected, Twitter changed its framework and eliminated the distinction, resulting in a single, directed relationship, follow, and a different mechanism for controlling who is notified about what posts.

By using the Twitter developer API[7], we fetched the social network of all users. We construct a directed graph $G(V, E)$, where V represents a set of users and E represents the set of "friend" relations. A directed edge *e* exists between two users *u* and *v* if user *u* declares *v* as a friend. There are a total of 87,897 distinct nodes with 829,053 friend relation between them. There are more nodes in this graph due to the fact that some users discovered though the link structure do not have any posts during the duration in which the data was collected. For each user, we also obtained their profile information and mapped their location to a geographic

[7] http://twitter.com/help/api

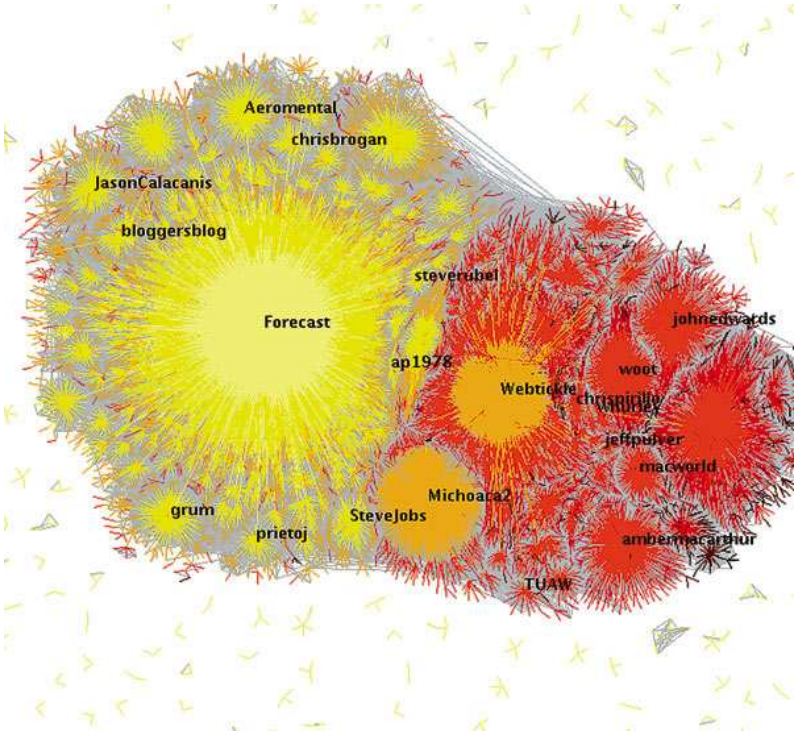


Figure 37: The graph constructed using the Large Graph Layout (LGL) tool. It consists of contacts from about 25K Twitter users. Notice that there is a link connecting two users if either one has the other as a friend and hence it is an undirected graph (of about 250K edges).

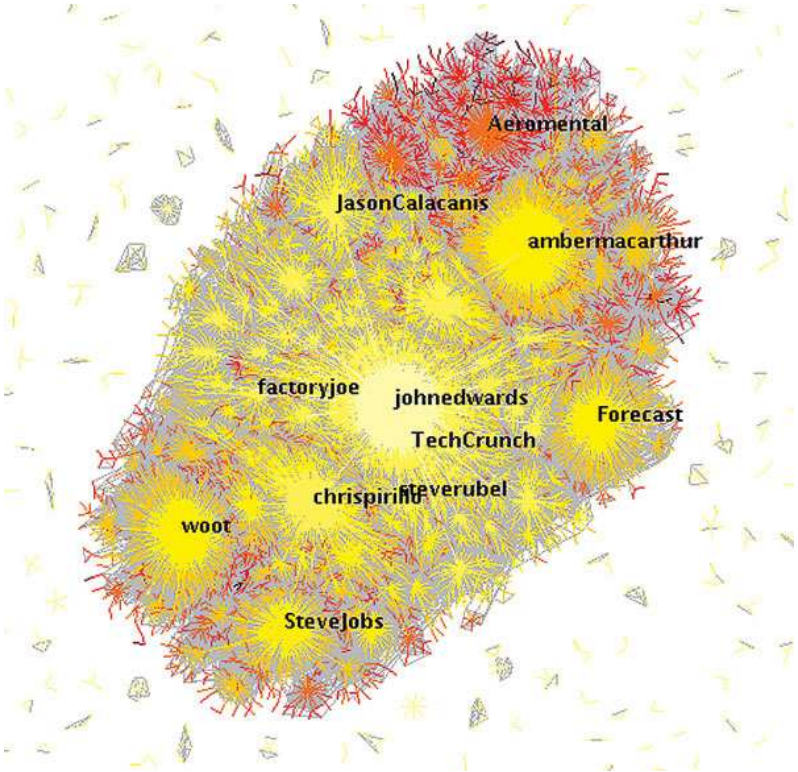


Figure 38: This graph is constructed using only users who are mutually acquainted. i.e. A knows B and also B knows A.

coordinate, details of which are provided in the following section.

Next we describe some of the characteristic properties of Twitter's social network, including its network topology, geographical distribution and common graph properties.

2. Microblogging Usage

Growth of Twitter

Since Twitter provides a sequential user and post identifier, we can estimate the growth rate of Twitter. Figure 39 shows the growth rate for users and Figure 40 shows the growth rate for posts in this collection. Since, we do not have access to historical data, we can only observe its growth for a two month time period. For each day we identify the maximum values for the user identifier and post identifier as provided by the Twitter API. By observing the change in these values, we can roughly estimate the growth of Twitter. It is interesting to note that even though Twitter launched in mid 2006, it really became popular soon after it won the South by SouthWest (SXSW) conference Web Awards[8] in March, 2007. Figure 39 shows the initial growth in users as a result of interest and publicity that Twitter generated at this conference. After this period, the rate at which new users are joining the network has declined. Despite the slow down, the number of new posts is constantly growing, approximately doubling every month indicating a steady base of users generating content.

Following Kolari et al. [109], we use the following definition of user activity and retention:

> **Definition.** A user is considered *active* during a week if he or she has posted at least one post during that week.
>
> **Definition.** An active user is considered *retained* for the given week, if he or she reposts at least once in the following X weeks.

Due to the short time period for which the data is available and the nature of microblogging, we chose as a value of X as a period of one week when computing user retention. Figure 41 shows the user activity and retention metrics for the duration of the data. About half of the users are active and of these, half of them repost in the following week. There is a lower activity recorded during the last week of the data due to the fact that updates from the public timeline are not available for two days during this period.

[8] http://2007.sxsw.com/

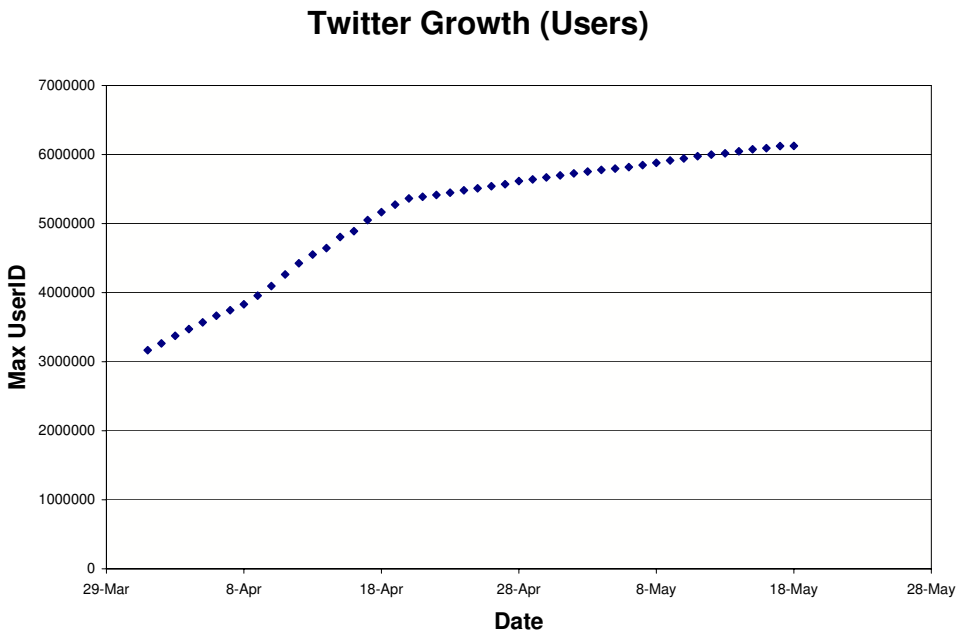


Figure 39: During the time we collected data Twitter was growing rapidly. This figure shows the maximum userid observed for each day in the dataset. After an initial period of interest around March 2007, the rate at which new users are joining Twitter slowed.

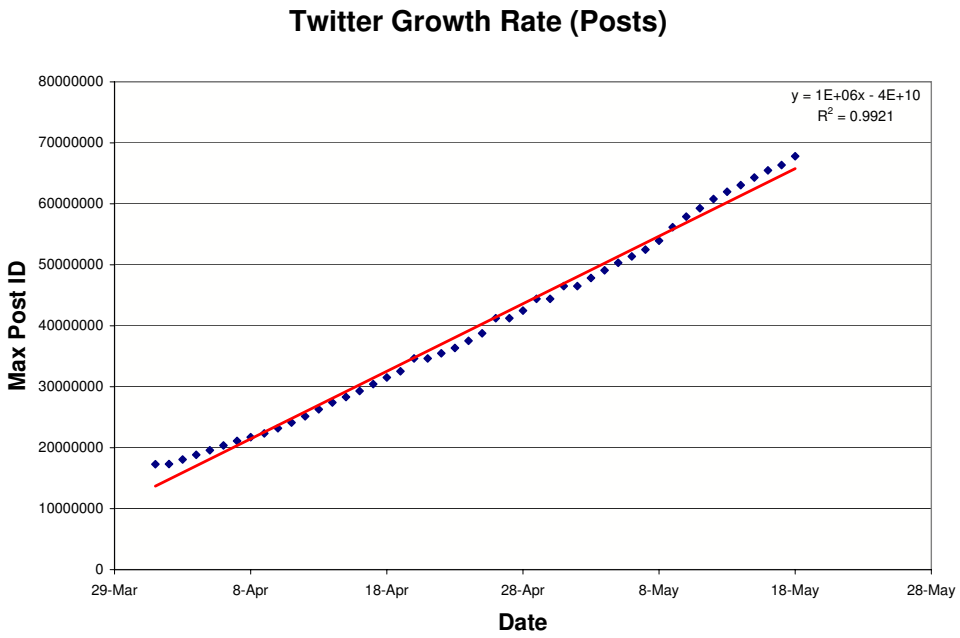


Figure 40: During the data collection period the number of posts increased at a steady rate even as the rate at which new users joined slowed. This figure shows the maximum post ID observed for each day in the dataset.

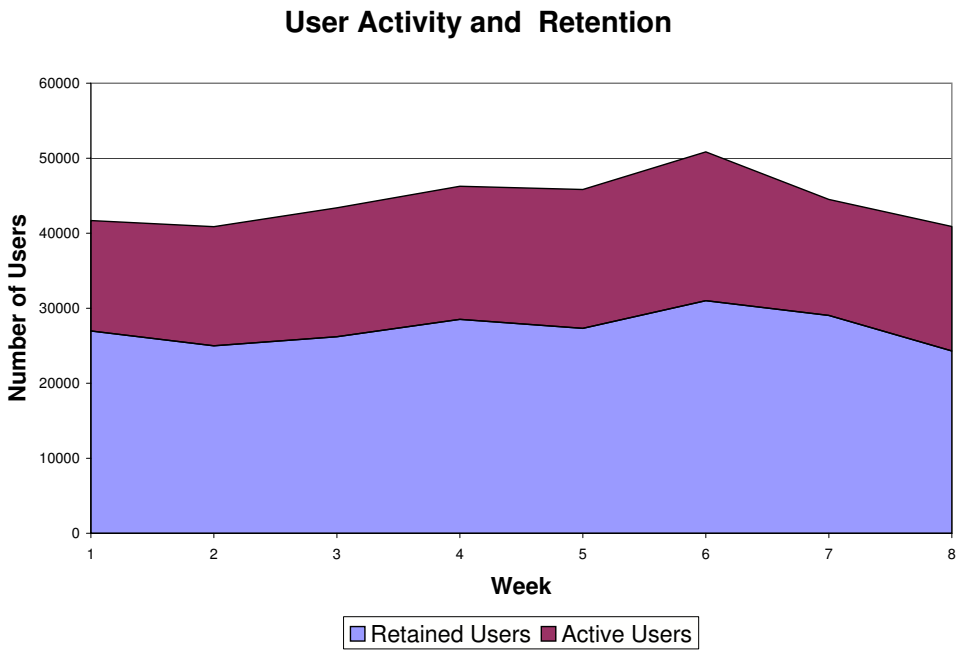


Figure 41: The number of active and retained users remained fairly constant during the time the data was collected.

Property	Twitter	WWE
Total Nodes	87897	143,736
Total Links	829247	707,761
Average Degree	18.86	4.924
Indegree Slope	-2.4	-2.38
Outdegree Slope	-2.4	NA
Degree correlation	0.59	NA
Diameter	6	12
Largest WCC size	81769	107,916
Largest SCC size	42900	13,393
Clustering Coefficient	0.106	0.0632
Reciprocity	0.58	0.0329

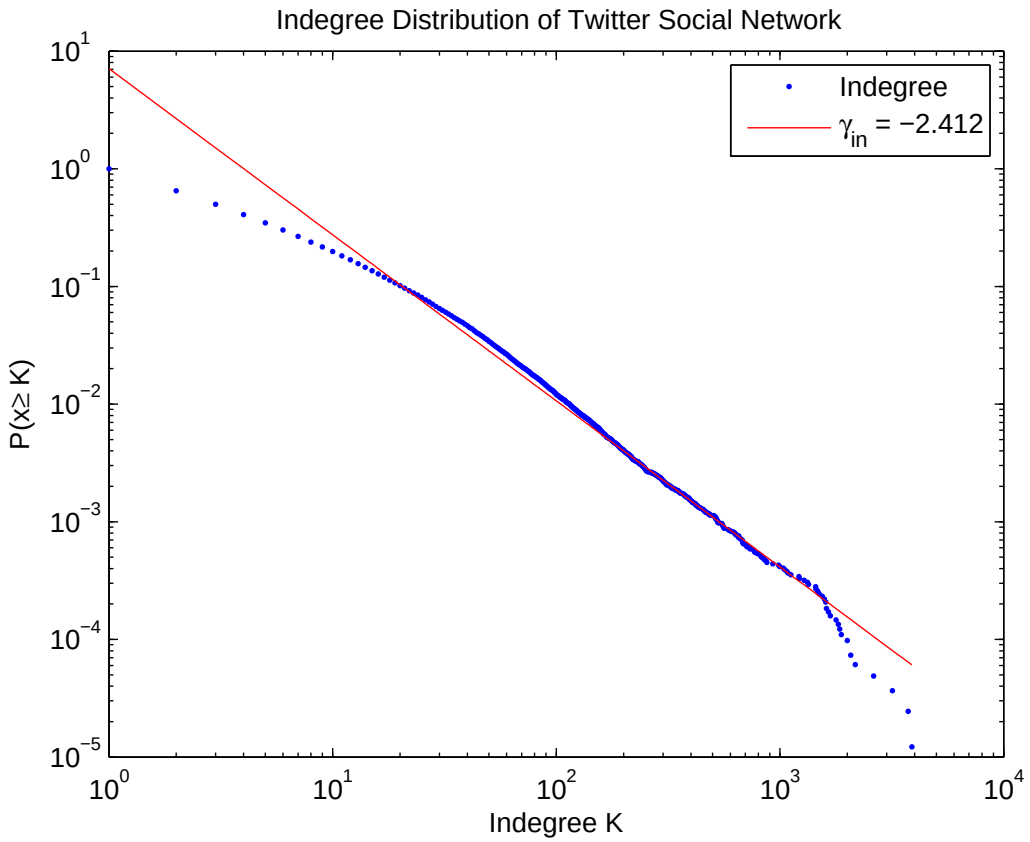
Table IV..10: This table shows the values of standard graph statistics for the Twitter social network.

Network Properties

The Web, Blogosphere, online social networks and human contact networks all belong to a class of "scale-free networks" [9] and exhibit a "small world phenomenon" [204]. It has been shown that many properties including the degree distributions on the Web follow a power law distribution [117, 20]. Recent studies have confirmed that some of these properties also hold true for the Blogosphere [185].

Table IV..10 describes some of the properties for Twitter's social network. We also compare these properties with the corresponding values for the Weblogging Ecosystems Workshop (WWE) collection [16] as reported by Shi et al. [185]. Their study shows a network with high degree correlation (also shown in Figure 43) and high reciprocity. This implies that there are a large number of mutual acquaintances in the graph. New Twitter users often initially join the network on invitation from friends. Further, new friends are added to the network by browsing through user profiles and adding other known acquaintances. High reciprocal links has also been observed in other online social networks like Livejournal [125]. Personal communication and contact network such as cell phone call graphs [152] also have high degree correlation. Figure 42 shows the cumulative degree distributions [157, 31] of Twitter's network. It is interesting to note that the slopes γ_{in} and γ_{out} are both approximately -2.4. This value for the power law exponent is similar to that found for the Web (typically -2.1 for indegree [44]) and Blogosphere (-2.38 for the WWE collection).

In terms of the degree distributions, Twitter's social network can thus be seen as being similar to the Web and Blogosphere, but in terms of reciprocity and degree correlation it is like a social network [125, 152].



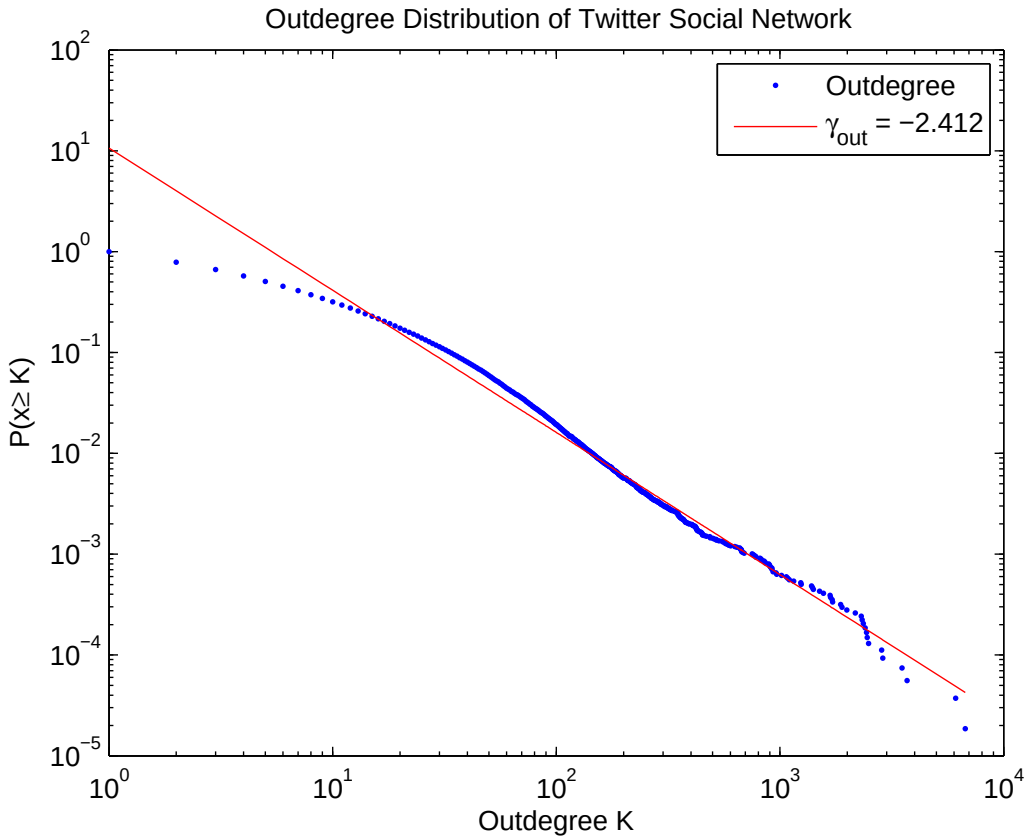


Figure 42: The Twitter social network has a power law exponent of about -2.4, which is similar to value exhibited by the Web and Blogosphere.

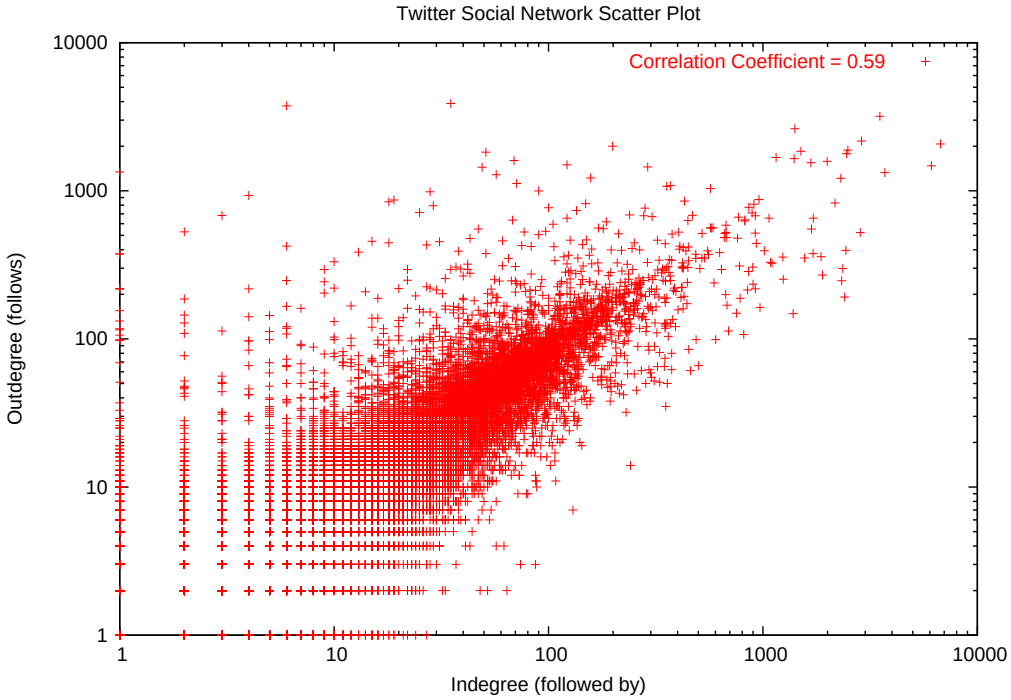


Figure 43: This scatter plot shows the correlation between the indegree and outdegree for Twitter users. A high degree correlation signifies that users who are followed by many people also have large number of friends.

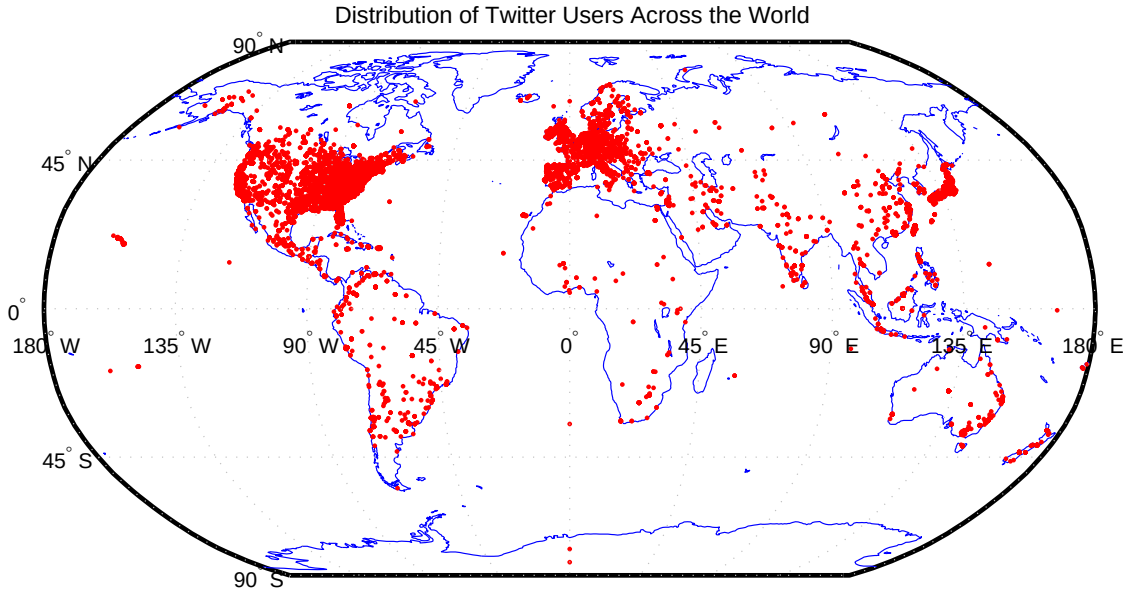


Figure 44: Although Twitter was launched in United States, it is popular across the world. This map shows the distribution of Twitter users in our dataset.

Continent	Number of Users
North America	21064
Europe	7442
Asia	6753
Oceania	910
South America	816
Africa	120
Others	78
Unknown	38994

Table IV..11: This table shows the geographical distribution of Twitter users, with North America, Europe and Asia exhibiting the highest adoption.

Geographical Distribution

Twitter provides limited profile information such as name, biographical sketch, timezone and location. For the 76 thousand users in our collection, slightly over half (about 39 thousand) had specified locations that could be parsed correctly and resolved to their respective latitude and longitudinal coordinates (using the Yahoo! Geocoding API[9]). Figure 44 and Table IV..11 show the geographical distribution of Twitter users and the number of users in each continent. Twitter is most popular in North America, Europe and Asia (mainly Japan). Tokyo, New York and San Francisco are the major cities where user adoption of Twitter is high [84].

Twitter's popularity is global and the social network of its users crosses continental boundaries. By mapping each user's latitude and longitude to a continent location we can extract the origin and destination location for every edge. Table IV..12 shows the distribution of friendship relations across major continents represented in the dataset. Oceania is used to represent Australia, New Zealand and other island nations. A significant portion (about 45%) of the social network still lies within North America. Moreover, there are more intra-continent links than across continents. This is consistent with observations that the probability of friendship between two users is inversely proportionate to their geographic proximity [125].

Table IV..13 compares some of the network properties across these three continents with most users: North America, Europe and Asia. For each continent the social network is extracted by considering only the subgraph where both the source and destination of the friendship relation belong to the same continent. Asian and European communities have a higher degree correlation and reciprocity than their North American counterparts. Language plays an important role is such social networks. Many users from Japan and Spanish speaking world connect with others who speak the same language. In general, users in Europe and Asia tend

[9] http://developer.yahoo.com/maps/

from-to	Asia	Europe	Oceania	N.A	S.A	Africa
Asia	**13.45**	0.64	0.10	**5.97**	0.005	0.01
Europe	0.53	**9.48**	0.25	6.16	0.17	0.02
Oceania	0.13	0.40	0.60	1.92	0.02	0.01
N.A	**5.19**	**5.46**	1.23	**45.60**	0.60	0.10
S.A	0.06	0.26	0.02	0.75	0.62	0.00
Africa	0.01	0.03	0.00	0.11	0.00	0.03

Table IV..12: This table shows the distribution of Twitter social network links across continents. Most of the social network lies within North America. (N.A = North America, S.A = South America)

Property	**N.A**	**Europe**	**Asia**
Total Nodes	16,998	5201	4886
Total Edges	205,197	42,664	60519
Average Degree	24.15	16.42	24.77
Degree Correlation	0.62	0.78	0.92
Clustering Coefficient	0.147	0.54	0.18
Percent Reciprocity	62.64	71.62	81.40

Table IV..13: Comparing the social network properties within continents shows that Europe and Asia have a higher reciprocity indicating closer ties in these social networks. (N.A = North America)

to have higher reciprocity and clustering coefficient values in their corresponding subgraphs.

3. Mining User Intention

Our analysis of user intention uses a two-level approach incorporating both HITS and community detection. First, we adapt the HITS algorithm [105] to find the hubs and authorities in the Twitter social network. An authority value for a person is the sum of the scaled hub values of her followers and her hub value is the sum of the scaled authority values of those she follows. Hubs and authorities have a mutually reinforcing property and are defined more formally as follows: $H(p)$ represents the hub value of the page p and $A(p)$ represents the authority value of a page p.

$$Authority(p) = \sum_{v \in S, v \to p} Hub(v)$$

And

$$Hub(p) = \sum_{u \in S, p \to u} Authority(u)$$

Table IV..14 shows a listing of Twitter users with the highest values as hubs and authorities. From this list, we can see that some users have high authority score, and also high hub score. For example, Scobleizer, JasonCalacanis, bloggersblog, and Webtickle who have many followers and friends in Twitter are located in this category. Some users with very high authority scores have relatively low hub score, such as Twitterrific,

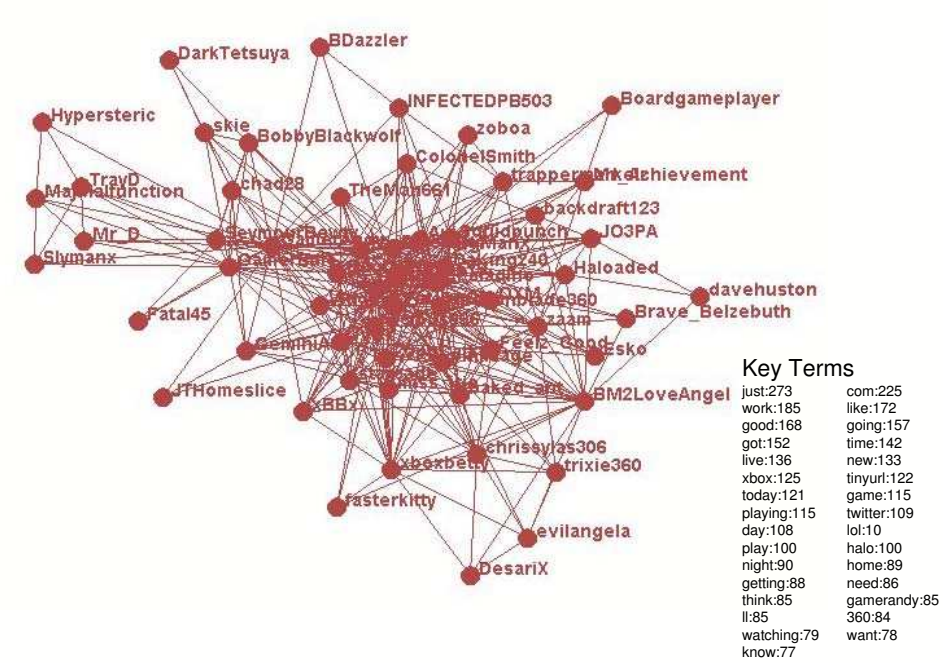


Figure 45: One of the user communities we discovered in the Twitter dataset is characterized by an interest in computer games, which is the major topic of the community members' posts. As is typical of other topic-based communities, the members also use Twitter to share their daily activities and experiences.

User	Authority	User	Hub
Scobleizer	0.002354	Webtickle	0.003655
Twitterrific	0.001765	Scobleizer	0.002338
ev	0.001652	dan7	0.002079
JasonCalacanis	0.001557	startupmeme	0.001906
springnet	0.001525	aidg	0.001734
bloggersblog	0.001506	lisaw	0.001701
chrispirillo	0.001503	bhartzer	0.001599
darthvader	0.001367	bloggersblog	0.001559
ambermacarthur	0.001348	JasonCalacanis	0.001534

Table IV..14: This table lists the Twitter users with the top hub and authority values computed from our dataset. Some of the top authorities are also popular bloggers. Top hubs include users like startupmeme and aidg which are microblogging versions of a blogs and other Web sites.

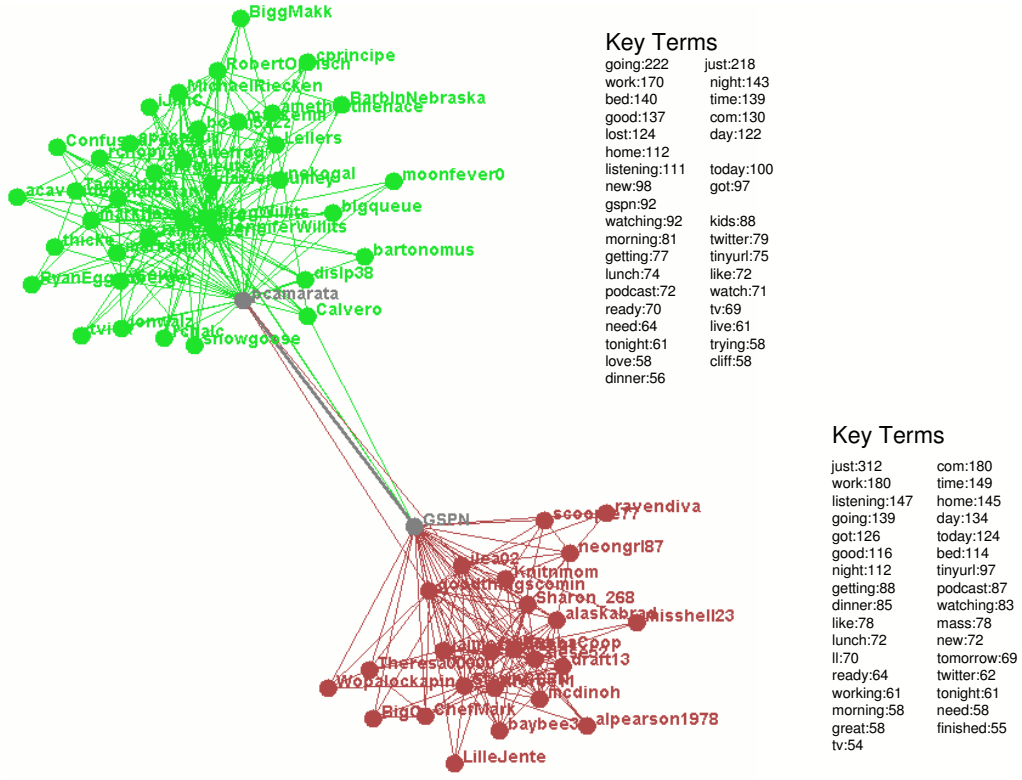


Figure 46: Our analysis revealed two Twitter communities in which podcasting was a dominant topic that are connected by two individuals. The communities differ in topic diversity, with the red community having a narrower focus.

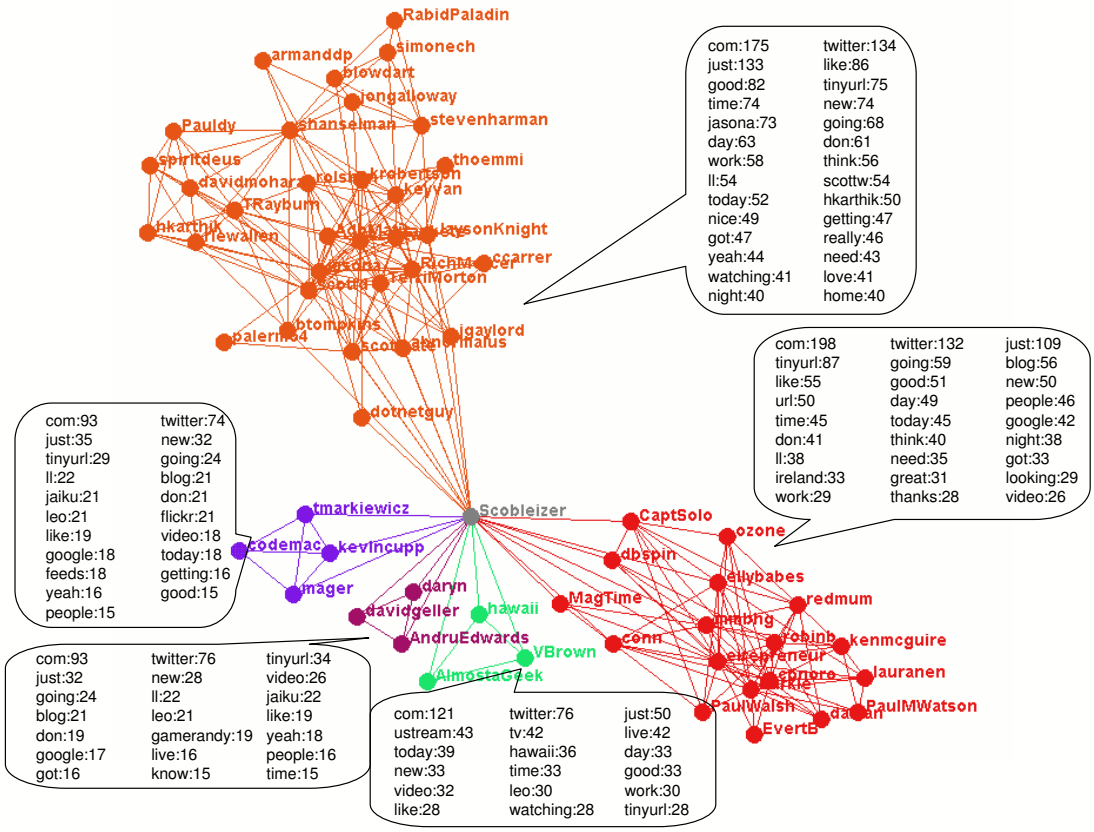


Figure 47: We identified five communities sharing technology as a dominant topic that are connected by a single user – *Scobleizer*, well known as a blogger specializing in technology.

ev, and springnet. They have many followers while less friends in Twitter, and thus are located in this category. Some other users with very high hub scores have relatively low authority scores, such as dan7, startupmeme, and aidg. They follow many other users while have less friends instead. Based on this rough categorization, we can see that user intention can be roughly categorized into these three types: information sharing, information seeking, and friendship-wise relationship.

After the hub/authority detection, we identify communities within friendship-wise relationships by only considering the bidirectional links where two users regard each other as friends. A community in a network can be defined as a group of nodes more densely connected to each other than to nodes outside the group. Often communities are topical or based on shared interests. To construct web communities Flake et al. [50] proposed a method using HITS and maximize flow/minimize cut to detect communities. In social network area, Newman and Girvan [55, 30] proposed a metric called modularity to measure the strength of the community structure. The intuition is that a good division of a network into communities is not merely to make the number of edges running between communities small; rather, the number of edges between groups is smaller than expected. Only if the number of between group edges is significantly lower than what would be expected purely by chance can we justifiably claim to have found significant community structure. Based on the modularity measure of the network, optimization algorithms are proposed to find good divisions of a network into communities by optimizing the modularity over possible divisions. Also, this optimization process can be related to the eigenvectors of matrices. However, in the above algorithms, each node has to belong to one community, while in real networks, communities often overlap. One person can serve a totally different functionality in different communities. In an extreme case, one user can serve as the information source in one community and the information seeker in another community.

In this section we describe some specific examples of how communities form in Twitter. Communities are the building blocks of any social network tools. Often the communities that develop are topical or based on shared interests. A community in a network is a group of nodes more densely connected to each other than to nodes outside the group. In naturally occurring networks, of course, communities often overlap.

People in friendship communities often know each other. Prompted by this intuition, we applied the Clique Percolation Method (CPM) [166, 40] to find overlapping communities in networks. The CPM is based on the observation that a typical member in a community is linked to many other members, but not necessarily to all other nodes in the same community. In CPM, the k-clique-communities are identified by looking for the unions of all k-cliques that can be reached from each other through a series of adjacent k-

cliques, where two k-cliques are said to be adjacent if they share k-1 nodes. This algorithm is suitable for detecting the dense communities in the network.

Here we give some specific examples of implicit Twitter communities and characterize the apparent intentions that their users have in posting. These "community intentions" can provide insight into why the communities emerge and the motivations users have in joining them Figure 45 illustrates a representative community with 58 users closely communicating with each other through Twitter service. The key terms they talk about include work, Xbox, game, and play. It looks like some users with gaming interests getting together to discuss the information about certain new products on this topic or sharing gaming experience. When we go to specific users website, we also find the following type of conversation.

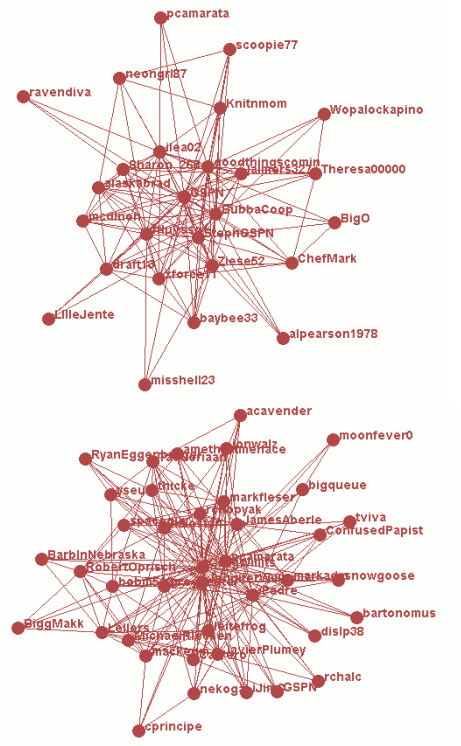
> *"BDazzler@Steve519 I don't know about the Jap PS3's. I think they have region encoding, so you'd only be able to play Jap games. Euro has no ps2 chip"or "BobbyBlackwolf Playing with the PS3 firmware update, can't get WMP11 to share MP4's and the PS3 won't play WMV's or AVI's...Fail."*

We also noticed that users in this community share with each other their personal feeling and daily life experiences in addition to comments on "gaming". Based on our study of the communities in Twitter dataset, we observed that this is a representative community in Twitter network: people in one community have certain common interests and they also share with each other about their personal feeling and daily experience.

Using the Clique Percolation Method we are able to find how communities are connected to each other by overlapping components. Figure 46 illustrates two communities with podcasting interests where the users *GSPN* and *pcamarata* connect these two communities. In GSPN's biographic sketch, he states that he is the producer of the *Generally Speaking Porkiest Network*[10]; while in pcamarata's bio, he mentioned he is a family man, a neurosurgeon, and a podcaster. By looking at the top key terms of these two communities, we can see that the focus of the green community is a little more diversified: people occasionally talk about podcasting, while the topic of the red community is a little more focused. In a sense, the red community is like a professional community of podcasting while the green one is a informal community about podcasting.

Figure 48 shows two example communities whose members tend to talk about their daily activities and thoughts. These discussions, while may seem mundane to most of us, will be of interest to close friends and family members. It is very similar to keeping in touch with friends and maintaining regular contact with them on chat or instant messengers. As pointed out by Reichelt [174], this use of microblogging can create a sense

[10]http://ravenscraft.org/gspn/home



Key Terms

going:222	just:218	work:170	night:143
bed:140	time:139	good:137	com:130
lost:124	day:122	home:112	listening:111
today:100	new:98	got:97	gspn:92
watching:92	kids:88	morning:81	twitter:79
getting:77	tinyurl:75	lunch:74	like:72
podcast:72	watch:71	ready:70	tv:69
need:64	live:61	tonight:61	trying:58
love:58	cliff:58	dinner:56	

Key Terms

years:11	script:10	pray:10	taquoriaan:10
lots:10	im:10	course:10	gone:10
wish:10	best:10	problem:10	11:10
ugh:10	leaving:10	lol:10	sick:10
gonna:10	15:10	probably:10	hungry:10
missed:10	link:10	starting:10	second:10
laptop:10	wall:10	finish:10	mom:10
20:10	forward:10	ate:10	saturday:10
couple:10	instead:10	high:10	hospital:10
started:10	listen:9	idea:9	place:9
needs:9	dad:9	played:9	recording:9
staying:9			

Figure 48: Analyzing the key terms in these two communities show that their members post updates that discuss the events of their daily activities.

of awareness and intimacy that transcends the constraints of space and time.

Figure 47 illustrates five communities connected by *Scobleizer*, who is well known as a blogger specializing in technology. People follow his posts to get technology news. People in different communities share different interests with *Scobleizer*. Specifically, the Twitter users *AndruEdwards*, *Scobleizer*, *daryn*, and *davidgeller* get together to share video related news. CaptSolo et al. have some interests on the topic of the Semantic Web. *AdoMatic* and others are engineers and have interests focused on computer programming and related topics.

Figure 46 shows how two seemingly unrelated communities can be connected to each other through a few weak ties [61]. While Twitter itself does not support any explicit communities, structures naturally emerge in the Twitter social network. Providing an easy way for users to find others in their implicit communities might be a useful service for systems like Twitter.

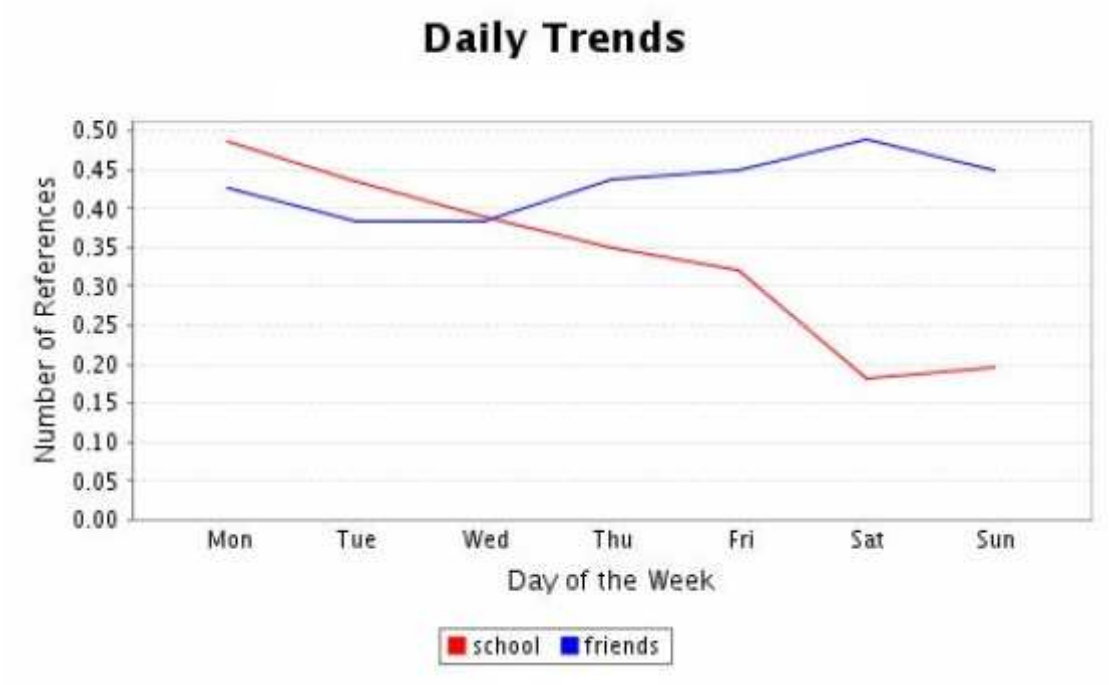


Figure 49: This graph shows the daily trends for terms “school”and “friends”. The term “school” is more frequent during the early week while “friends” take over during the weekend.

	Day	Other Days	Total
Frequency of word	a	b	a+b
Frequency of other words	c-a	d-b	c+d-a-b
Total	c	d	c+d

Studying intentions at a community level, we observe users participate in communities that share similar interests. Individuals may have different intentions in becoming a part these implicit communities. While some act as information providers, others are merely looking for new and interesting information. Next, we analyze aggregate trends across users spread over many communities, we can identify certain distinct themes. Often there are recurring patterns in word usages. Such patterns may be observed over a day or a week. For example Figure 49 shows the trends for the terms “friends”and “school” in the entire corpus. While school is of interest during weekdays, the frequency of the friends term increases during the week and dominates on the weekends.

The log-likelihood ratio is used to determine terms that are of significant importance for a given day of the week. Using a technique described by Rayson and Garside [173], we create a contingency table of term frequencies for each of the day of the week and the remaining days in the week.

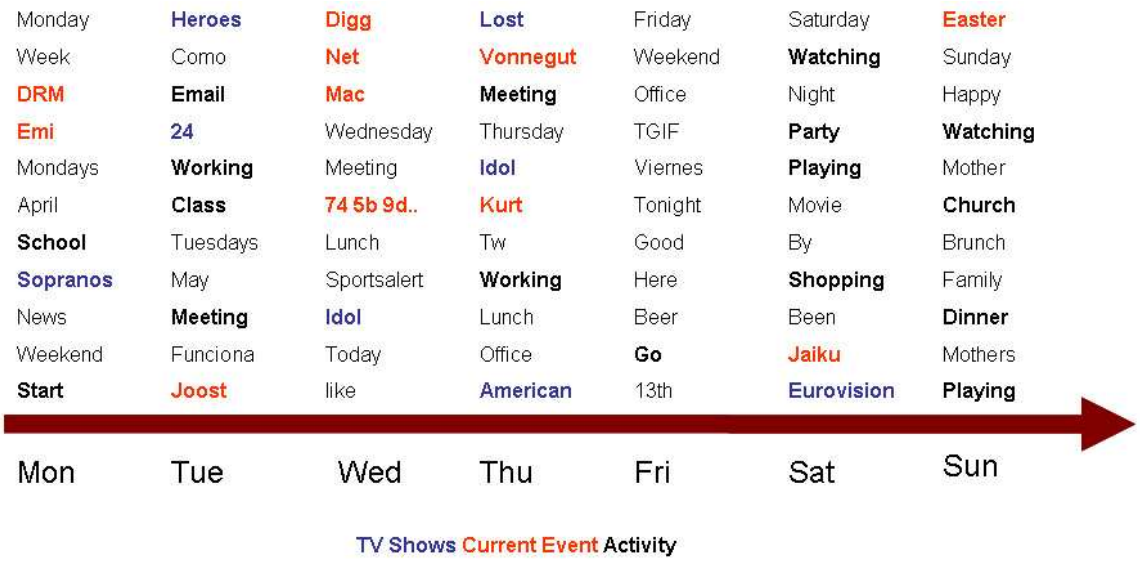


Figure 50: Distinctive terms for each day of the week ranked using Log-likelihood ratio.

Comparing the terms that occur on a given day with the histogram of terms for the rest of the week, we find the most descriptive terms. The log-likelihood score is calculated as follows:

$$LL = 2 * (a * log(\frac{a}{E1}) + b * log(\frac{b}{E2})) \quad \text{(IV..13)}$$

where $E1 = c * \frac{a+b}{c+d}$ and $E2 = d * \frac{a+b}{c+d}$

Figure 50 shows the most descriptive terms for each day of the week. Some of the extracted terms correspond to recurring events and activities significant for a particular day of the week for example “school”or “party”. Other terms are related to current events like “Easter”and “EMI”.

4. Conclusions

This section presents a brief taxonomy of user intentions in the Twitter microblogging community. The apparent intention of a Twitter post was determined manually by the first author. Each post was read and categorized. Posts that were highly ambiguous or for which the author could not make a judgement were placed in the category UNKNOWN. Based on this analysis we have found the following as the main user intentions in Twitter posts.

- *Daily Chatter.* Most posts on Twitter talk about daily routine or what people are currently doing. This is the largest and most common user of Twitter.

- *Conversations.* In Twitter, since there is no direct way for people to comment or reply to their friend's posts, early adopters started using the @ symbol followed by a username for replies. About one eighth of all posts in the collection contain a conversation and this form of communication was used by almost 21% of users in the collection.

- *Sharing information/URLs.* About 13% of all the posts in the collection contain some URL in them. Due to the small character limit of Twitter updates, a URL shortening service like TinyURL[11] is frequently used to make this feature feasible.

- *Reporting news.* Many users report latest news or comment about current events on Twitter. Some automated users or agents post updates like weather reports and new stories from RSS feeds. This is an interesting application of Twitter that has evolved due to easy access to the developer API.

Using the link structure, following are the main categories of users on Twitter:

- *Information Source.* An Twitter user who is an information source is also a hub and has a large number of followers. This user may post updates on regular intervals or infrequently. Despite infrequent updates, certain users have a large number of followers due to the valuable nature of their updates. Some of the information sources were also found to be automated tools posting news and other useful information on Twitter.

- *Friends.* Most relationships fall into this broad category. There are many sub-categories of friendships on Twitter. For example a user may have friends, family and co-workers on their friend or follower lists. Sometimes unfamiliar users may also add someone as a friend.

- *Information Seeker.* An information seeker is a person who might post rarely, but follows other users regularly.

Our study has revealed different motivations and utilities of microblogging platforms. A single user may have multiple intentions or may even serve different roles in different communities. For example, there may be posts meant to update your personal network on a holiday plan or a post to share an interesting link with co-workers. Multiple user intentions have led to some users feeling overwhelmed by microblogging services [120]. Based on our analysis of user intentions, we believe that the ability to categorize friends into groups

[11] http://www.tinyurl.com

(e.g. family, co-workers) would greatly benefit the adoption of microblogging platforms. In addition features that could help facilitate conversations and sharing news would be beneficial.

In this study we have analyzed a large social network in a new form of social media known as microblogging. Such networks were found to have a high degree correlation and reciprocity, indicating close mutual acquaintances among users. While determining an individual user's intention in using such applications is challenging, by analyzing the aggregate behavior across communities of users, we can describe the community intention. Understanding these intentions and learning *how* and *why* people use such tools can be helpful in improving them and adding new features that would retain more users.

We collected two months of data from the Twitter microblogging system, including information on users, their social networks and posts. We identified different types of user intentions and studied the community structures. Our ongoing work includes the development of automated approaches of detecting user intentions with related community structures and the design of efficient techniques to extract community structures from very large social networks [87].

Chapter V.

INFLUENCE AND TRUST

These days Social Media tools like forums, wikis and blogs, in particular, are playing a notable role in influencing the buying patterns of consumers. Often a buyer looks for opinions, user experiences and reviews on such sources before purchasing a product. Detecting influential nodes and opinion leaders and understanding their role in how people perceive and adopt a product or service provides a powerful tool for marketing, advertising and business intelligence. This requires new algorithms that build on social network analysis, community detection and opinion extraction.

In this chapter, we discuss two approaches to detecting *influential* feeds. The first approach is based on mining the wisdom of the crowds to measure the importance of a feed in a given topic. The idea is that if a large number of users subscribe to Dailykos, and categorize it under "politics", for example, then it is significantly influential about this subject. Aggregated feed readership across several thousands of users provides sufficient evidence to categorize and rank blogs and feeds in various topics. Moreover, it also provides a means to recommend new feeds based on the similarity of a new user's subscriptions to other users like her.

Research in the area of information propagation was inspired by a large body of work in disease and epidemic propagation. One way of modeling the spread of ideas, memes and topics on the blogosphere is using epidemic propagation. In such models, if a sufficient fraction of a node's immediate neighbors have adopted an idea, the node will also be *infected* by this topic. In the second section of this chapter, we discus how influential nodes can be identified using epidemic propagation models. These techniques have been found to be effective in performing analysis at an aggregate level and to identify key individuals who play an important role in propagating information. However, influence on the Web is often a function of topic.

A blog like Daily Kos that is influential in politics is less likely to have an impact on the technology related blogs. Similarly, Techcrunch, an extremely popular technology blog might not be influential when it comes to politics. We propose the notion of 'topical influence' and extend existing techniques to make them topic sensitive.

A. Finding High Quality Feeds

Blogs have become a means by which new ideas and information spread rapidly on the Web. They discuss the latest trends and react to events as they unfold around the world. Protocols such as RSS, ATOM and OPML and services such as Blog search engines and ping servers have made it much easier to share information online. RSS and ATOM are XML-based file formats used for syndication. Outline Processor Markup Language (OPML) is a popular XML based format used to share an outline of the feed subscriptions.

Today, the feed infrastructure provided by RSS and ATOM is being used to serve a wide variety of online content, including blogs, wikis, mainstream media, and search results. All support different forms of syndication. Users can subscribe to feeds using reader such as Bloglines[1], Google Reader[2], News Gator[3], etc. Typically, a user adds a feed in a feed reader when she came across it (perhaps, by chance) as a reference on another blog. This is not always the best way to find good feeds.

A number of blog search engines and some hand-crafted directories try to provide a high quality index of feeds. Blog search engines such as Technorati[4] have introduced new features enabling people to find authoritative feeds on a given topic. The blog finder feature works by relying on the author of the blog to provide the tags. Further it ranks the blogs based on the number of inlinks. These problems make it insufficient in terms of finding topically authoritative blogs.

Hand-crafted directories have the disadvantage that they are based on the decision of the site creator. Additionally, there are only a limited set of sites that one can categorize manually. Recent efforts in tackling these problems have resulted in *Share your OPML*[5], a site where you can upload an OPML feed to share it with other users. This is a good first step but the service still does not provide the capability of finding good feeds topically.

An alternative is to search for blogs by querying blog search engines with generic keywords related to

[1] http://www.bloglines.com
[2] http://www.google.com/reader
[3] http://www.newsgator.com
[4] http://www.technorati.com
[5] http://share.opml.org/

the topic. However, blog search engines present results based on the *freshness*. Query results are typically ranked by a combination of how well the blog post content matches the query and how recent it is. Measures of the blog's authority, if they are used, are mostly based on the number of inlinks. These factors make it infeasible to search for new feeds by querying blog search engines. Moreover, this can sometimes be slightly misleading since a single post from a popular blogger on any topic may make him the top-most blog for that topic, even if his blog has little to do with the given subject.

Finding high-quality and topically authoritative feeds remains a challenge. In this section, we study the feed subscriptions of a large sample of Bloglines publicly listed users. Using this data, we first characterize the general feed usage patterns. Next, we identify the feeds that are popular for a given topic using folders names as an approximation for a topic. By merging related folders we can create a more appropriate and compact set of topics. Finally, we discuss some of the preliminary results in using this approach in support of a number of blog-related applications: feed browsing, feed recommendations, and searching for influential blogs in a different dataset.

1. Related Work

Blog hosting tools, search services and Web 2.0 sites such as Flickr[6] and del.icio.us[7] have popularized the use of tags. Tags provide a simple scheme that helps people organize and manage their data. Tags across all the users, collectively, are termed as a *folksonomy* [170], a recent term used to describe this type of user-generated content. Tags are like keywords used in the META tag of HTML. Adam Mathes [134] suggests that there are two reasons why people may use tags: to classify information for themselves or to help a community of users.

Brooks and Montanez [21] have studied the phenomenon of user-generated tags to evaluate effectiveness of tagging. Their study presents an analysis of the 250 most frequently used Technorati tags. Brooks et al. find that tagging can be helpful for grouping related items together but does not perform as well as text clustering. A text-based hierarchical clustering was used to group related tags together. We study similar problems, with the aim of finding important feeds for a topic. By using a dataset that is based on feed subscriptions rather than text in individual posts, we can group similar feeds together. Another study of a social bookmark tool, del.ici.ous, by Cattuto et al.[23], presents an analysis of collaborative tagging. Their research indicates that a common vocabulary emerges across user-generated tags and they also provide a stochastic model that

[6] http://www.flickr.com
[7] http://del.icio.us

approximates this behavior.

Shen and Wu [182] treat tags as nodes and the presence of multiple tags for a document as a link between the tags. According to Shen, the network structure and properties of such a graph resemble that of a scale-free network. In our analysis, we study the different usage and subscription characteristics of feeds and find that some of these features also follow a power law distribution. While such distributions would not be surprising anymore, it is interesting to note that while the total number of blogs are increasing, the feeds that matter are actually just a small portion of the Blogosphere.

Guy and Tonkin [70] discuss the issue of cleaning up the tag space. Their study of del.icio.us and Flickr tags found that a significant number of tags are misspelled. User enforced hierarchies created with tags separated by special characters accounts for a portion of the tag space. The biggest advantage of folksonomies is that it gives people the flexibility to label content using any terms that they find appropriate. Enforcing a set of rules or suggesting tag selection guidelines is helpful but not easy to implement. In this paper we propose an alternative, where variations of tag or folder name usage can automatically be inferred through merging related tags. This allows users to continue creating their own tags, while improving topical relevance of systems using this information.

An alternative suggested to improve the quality of tagging is AutoTagging [146]. Social bookmark tools like del.icio.us already provide suggestions for tagging a URL based on terms used to describe the same link by other users in the system. AutoTagging is a collaborative filtering based recommendation system for suggesting appropriate tags. The suggested tags are based on tags used for other posts that are similar in content. This work does not directly address AutoTagging but we describe a similar approach for a slightly different motivation - finding topically authoritative feeds and recommending new feeds based on tag usage similarity.

Dubinko et al. [46] describe tag visualization techniques by using Flickr tags. Their work concentrates on automatically discovering tags that are most ‘interesting’ for a particular time period. By visualizing these on a timeline they provide a tool for exploring the usage and evolution of tags on Flickr. In this work we take only a static view of feed subscriptions and folder usage. Feed subscriptions unlike flickr or technorati tag clouds evolve rather slowly and hence taking a static view of the data is not too unrealistic.

Marlow [133] compares blogroll links and permalinks (URLs of specific blog post) as features to determine authority and influence on the Blogosphere. The study suggests that permalink citations can approximate influence. Present blog search engines indeed use permalink citations or inlinks to a blog as a measure

Domain	Percentage	domain	Percentage
blogspot	24.36	hatena	1.07
livejournal	3.81	topix	0.89
flickr	2.89	technorati	0.75
msn	1.73	wretch	0.56
typepad	1.73	exblog	0.54
yahoo	1.71	wordpress	0.47
xanga	1.43	msdn	0.45
icio	1.24	blogs	0.45
google	1.22	rest	53.60
livedoor	1.10		

Table V..1: The distribution of domains in the Bloglines dataset

of authority. The disadvantage is that such measures do not work well when the goal is to find authoritative blogs in a particular topic. In our approach, we treat folder names as an approximation of topic and number of subscribers as an indication of the authority. We find that such measures are effective in finding topically authoritative blogs.

2. Dataset Description

Bloglines is a popular feed reader service. Using this tool makes it easy to monitor a large number of RSS feeds. Once a user subscribes to a set of feeds, this service monitors the subscriptions and allows the user to view unread posts from their subscribed feeds. The simple user interface and convenient feed monitoring ability have made Bloglines an extremely popular feed reader. Bloglines provides a feature wherein users may choose to share their subscriptions. We conduct a study of the publicly listed OPML feeds from 83,204 users consisting of a total of 2,786,687 subscriptions of which 496,879 are unique. These are essentially the *"feeds that matter"* [119] since they are feeds that people have actually subscribed to. Table V..1 shows the distribution of the top domains in the Blogines dataset. In particular, there are a number of users who subscribe to Web 2.0 sites and dynamically generated RSS feeds over customized queries. It was also interesting to note that even though Blogspot has had serious splog issues [168, 110], based on the Bloglines dataset, it still contributes to a significant portion of the feeds that really matter on the Blogosphere.

According to Bloglines/Ask in July 2005 there were about 1.12 Million feeds that really matter, which is based on the feeds subscribed by all the users on Bloglines. A study of the feeds on Bloglines by McEvoy [136] in April 2005 showed that there were about 32,415 public subscribers and their feeds accounted for 1,059,140 public feed subscriptions. We collected similar data of the publicly listed users on Bloglines. From

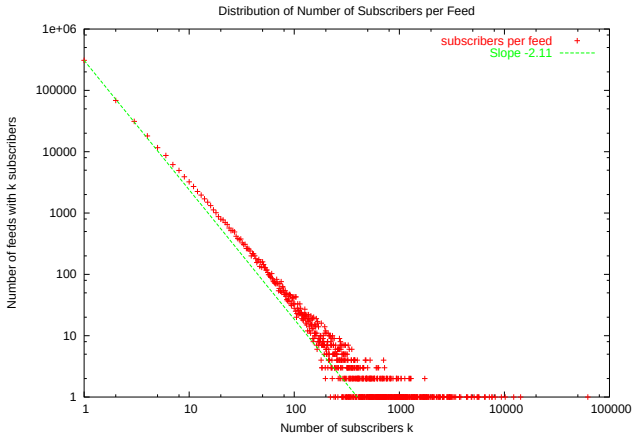


Figure 51: The number of subscribers for feeds follows a power law distribution.

last year, the number of publicly listed subscribers had increased to 83,204 users (2.5 times that of last year) and there were 1,833,913 listed feeds (1.7 times) on the Bloglines site. Hence, even though the Blogosphere is almost doubling every six months [188], we found that the number of feeds that *"really matter"* doubles roughly every year. Inspite of this, popularly subscribed feeds are still only a small fraction of the entire Blogosphere. Following is a description of some of the usage patterns and interesting statistics obtained from our analysis.

Figure 2. shows the distribution of the number of subscribers for 496,879 unique feeds across 83,204 users. This graph indicates a typical power law behavior with a few feeds having a large number of subscribers while most having a small number of subscribers. The exponent of the curve was found to be about -2.1 which is typical in scale-free systems and WWW [4]. While the presence of a power law distribution across feed subscription is expected, it is interesting to observe that even across a large sample of users, the number of unique feeds subscribed is fairly small in comparison to the 53 Million blogs on the Blogosphere [188].

Next, we analyzed the number of feeds subscribed per user. The number of subscribers for a feed is an indication of its authority and influence over its audience. Figure 52 depicts the distibution of the number of feeds subscribed across all users. Almost 90% of the users have less than 100 subscriptions. It is possible that for most users there is an inherent limit on the amount of information that they can keep track of at any given time. This limits their attention and hence the number of feeds that they typically subscribe to.

Bloglines has a feature by which a user may organize their feeds into different folders. While only some (26,2436 or about 35%) of the public subscribers use folders, it provides a user generated categorization of feeds. Figure 53 shows the histogram of folder usage across all users. While the folder organization is not

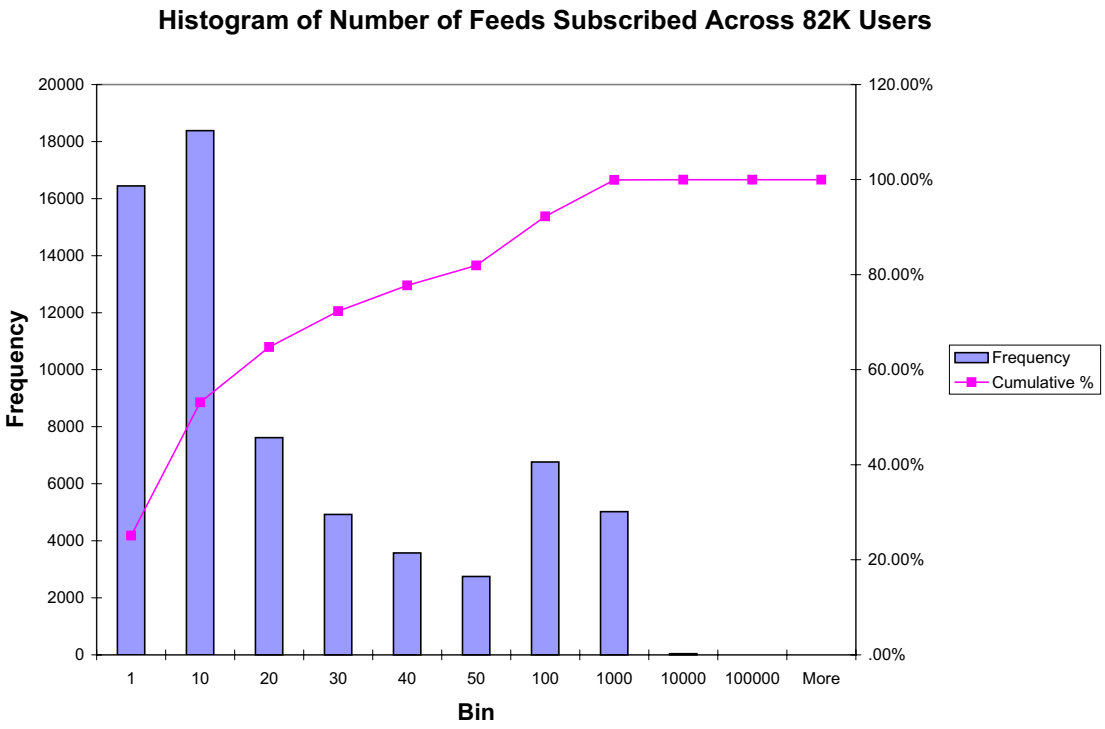


Figure 52: The Histogram of feeds subscribed per user

a very commonly used feature, most users who do use them have a relatively small number of folders. A vast majority of users had only one folder - named 'subscriptions' folder created by default for all users. Almost 90% of users have less than 10 folders and only roughly 100 users had more than 100 folders. Figure 54 shows a scatter plot of the number of folders compared to the number of feeds subscribed across all users. Although there is a very high variance, it can be observed from this graph that as the number of feeds subscribed increase, users generally organize them into greater number of folders.

Figure 55 shows the folder usage across all subscriptions. Each folder is ranked by the number of distinct feeds that have been categorized into that particular folder. It can be observed that the highly ranked folders are also those that are used by many subscribers. Thus the usage pattern suggests a consensus based on a folksonomy [8] emerges and a common vocabulary is being used to tag the feeds.

[8] http://en.wikipedia.org/wiki/Folksonomy

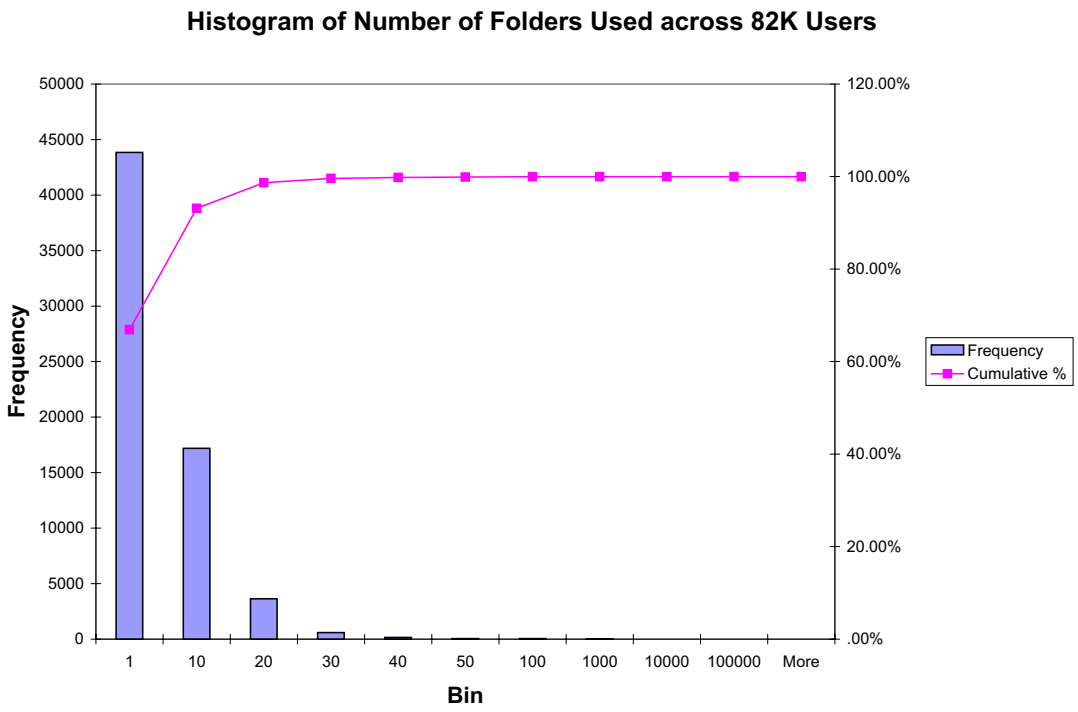


Figure 53: The histogram of folder usage across all users

3. Clustering Related Topics

Folder names can be treated as an approximation of a topic. Folder names in Bloglines are used in a way that is similar to Folksonomies on the web. As shown in Figure 56, by aggregating folders across all users, we can generate a tag cloud that shows the relative popularity and spread of various topics across Bloglines users. The tag cloud shown here is based on the top 200 folders. Note that the tag cloud contains terms such as 'humor' and 'humour', etc. These terms represent variations in which different users label feeds. By merging folder names that are *'related'* we can generate a more appropriate and compact representation of the tag cloud. Automatic techniques for inferring concept hierarchies using clustering [179] WordNet [142] and other statistical methods [58] have been found to be effective in finding relationships between topics.

The following section describes an approach used to merge related folders together. We were first tempted to use a morphologicial approach – merging the *blog* and blogs categories, for example. However, we soon discovered that folders with lexically similar names might actually represent different categorization needs of the users. For example, the folder 'Podcasting' consists of feeds that talk about how to podcast and provide tools. On the other hand 'Podcasts' refers to feeds containing actual podcasts. Other examples include 'Music' vs. 'Musica' (a topic with Spanish music blogs).

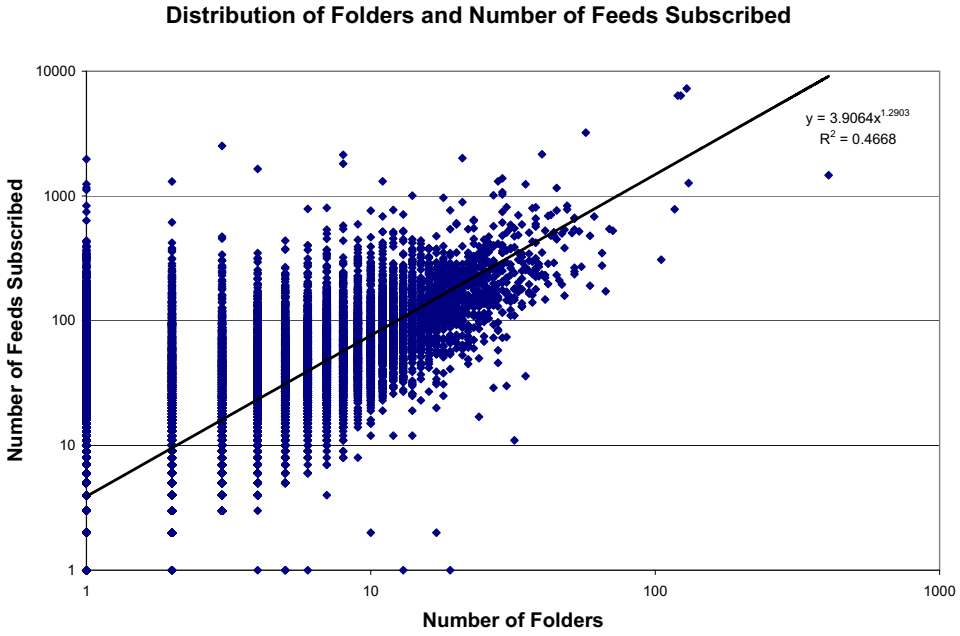


Figure 54: Scatter plot showing the relation between the number of folders and number of feeds subscribed. Note: This includes the feeds subscribed under the default folder labeled ‘Subscriptions’

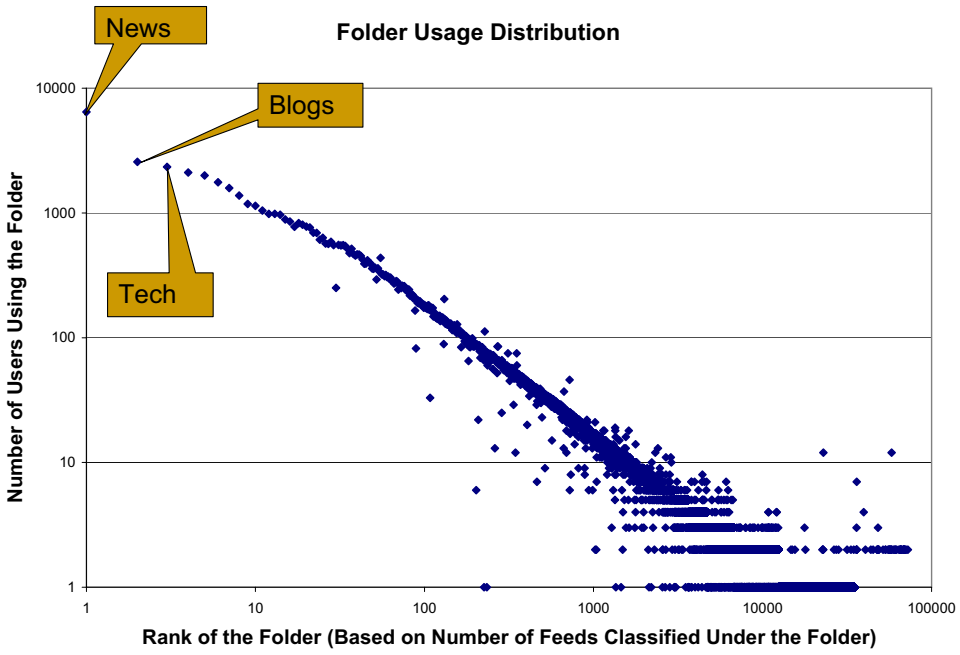


Figure 55: Folder usage distribution. The rank of the folder is computed by the number of distinct feeds that are categorized under that folder name.

Before Merge

After Merge

Figure 56: The tag cloud generated from the top 200 Folders before and after merging related folders. The size of the word is scaled to indicate how many users use the folder name.

For each folder we construct a vector containing the feeds that have been categorized under that folder name and their corresponding counts. At this step we take only the top 100 most frequently occurring feeds per folder. This threshold was heuristically determined. Some folders, such as ‘friends’, were observed to consist of a large set of feeds for each of which there are only a handful of subscribers. On the other hand extremely popular folders like ‘politics’ contained a number of feeds that have many subscribers.

Two cases need to be considered for computing folder similarity: first is the case where feeds in one folder may either partially or completely subsume feeds present in another folder. Complete subsumption indicates that there is a broader category and the larger folder is more general while partial subsumption indicates that the two categories are related. For example the folder ‘news’ subsumes a number of folders that are more specific, such as ‘tech news’,‘IT news’, ‘general news’, etc. For detecting the topics, it suffices to put these into a single category titled ‘news’. To compute subsumption we first find an overlap factor. For all folder pairs i,j we maintain a score of the overlap of feeds in folder j with feeds in folder i as follows:

$$overlap = \frac{matches}{size_j}$$

Folder similarity can be described in terms of the feeds that are contained in the folders. Two folder names are considered to be similar if they contain similar feeds in them. For each pair of folder names we compute the cosine similarity as follows:

$$cos(i,j) = \frac{\sum_k W_{i,k} W_{j,k}}{\sqrt{\sum_k W_{i,k}^2 * \sum_k W_{j,k}^2}}$$

The weight $W_{i,k}$ is determined by a TFIDF score for each feed in the vector. The weights are computed using the following formula:

$$W_{folder}(feed) = freq_{folder}(feed) * \log(\frac{folderCount}{|foldersContaining(feed)|})$$

First we start by ranking the folders based on the number of users using the folder. Next we go through this ranked list and merge related folders together. A lower ranked folder is merged into a higher ranked folder if $overlap > \theta$ or $cosine > \delta$. These thresholds were empirically set to 0.4. Setting it to smaller values leads to lowering the criteria for grouping, resulting in fewer topics and higher values resulted in fewer groupings resulting in more topics. Table V..2 shows examples of folders names with the corresponding merged folders.

Folder	Merged Folders
comics	fun, humor, funny, humour, cartoons, fun stuff, webcomics, comix, comic strips
music	mp3, mp3 blogs
politics	political, political blogs
design	web design, web, web development, webdesign, webdev, css, web dev,web standards
programming	development, dev, technical, software development, code
culture	miscellaneous, random, misc. , interesting
productivity	gtd, lifehacks, getting things done

Table V..2: Example folders with corresponding merged sub-folders

1	http://www.talkingpointsmemo.com
2	http://www.dailykos.com
3	http://atrios.blogspot.com
4	http://www.washingtonmonthly.com
5	http://www.wonkette.com
6	http://instapundit.com
7	http://www.juancole.com
8	http://powerlineblog.com
9	http://americablog.blogspot.com
10	http://www.crooksandliars.com

Table V..3: The Feeds That Matter for **'Politics'**

Once the merging of related folders is completed, a list of feeds relevant and authoritaive for a topic can be created. This task is similar to automatic resource compilation [24] which aims to find authoritative Web resources by analyzing the link structure. In our approach we say that a feed is topically relevant and authoritative if many users have categorized it under the given folder name. After merging related folders together, the total number of times each of the feeds appears across all the merged folders is added to obtain the final ranking. Tables V..4 and V..3 provide examples of feeds that matter for "Photography" and "Politics" that were found using this technique.

1	http://wvs.topleftpixel.com
2	http://blog.flickr.com/flickrblog/
3	http://www.flickr.com/recent_comments.gne
4	http://www.east3rd.com
5	http://www.durhamtownship.com
6	http://www.digitalcamerawebsites.com
7	http://groundglass.ca/
8	http://www.photographica.org/
9	http://chromogenic.net/
10	http://www.backfocus.info/

Table V..4: The Feeds That Matter **'Photography'**

4. Applications

The availablility of subscription information provides an opportunity to cluster related blogs together based on how certain blogs are subscribed together by different users as well as how these blogs are grouped or organized under different folder names. For each 'topic' we can also derive the list of blogs that are most widely read. Readership can be a useful metric in measuring influence of blogs for a particular topic [93]. In this section, we present two more use cases where the knowledge of such subscription information can be helpful.

Feed Recommender

Folder similarity allows us to compare how related two folder vectors are based on the feeds that occur in them. Feed similarity can be defined in a similar manner: two feeds are similar if they often co-occur under similar folders. Note that this definition of feed similarity does not use the textual content of the feed but is entirely based on the subscription data. This gives us an ability to compare two feeds and recommend new feeds that are like a given feed. For each feed there is a folder vector that maintains a count of the number of times the feed has been categorized under a folder name. For a pair of feeds i,j feed similarity is defined as:

$$cos(i,j) = \frac{\sum_k W_{i,k} W_{j,k}}{\sqrt{\sum_k W_{i,k}^2 * \sum_k W_{j,k}^2}}$$

The weight $W_{i,k}$ is determined by a TFIDF score for each folder in the feed vector. The weights are computed using the following formula:

$$W_{feed}(folder) = freq_{feed}(folder) * log(\frac{feedCount}{|feedsLabeled(folder)|})$$

The feed similarity measure could be further improved by using folder similarity as computed in the previous section. Two feeds are similar if they occur in similar folders (rather than identical folders).

FTM! Feeds That Matter

FTM![9] is a site that was implemented out of a need to find a high quality listing or index of *topical* blogs and feeds. This site is based on the Bloglines dataset described in this paper and implements the algorithms presented here for merging folders and providing recommendations. For example if the user was interested in

[9] http://ftm.umbc.edu/

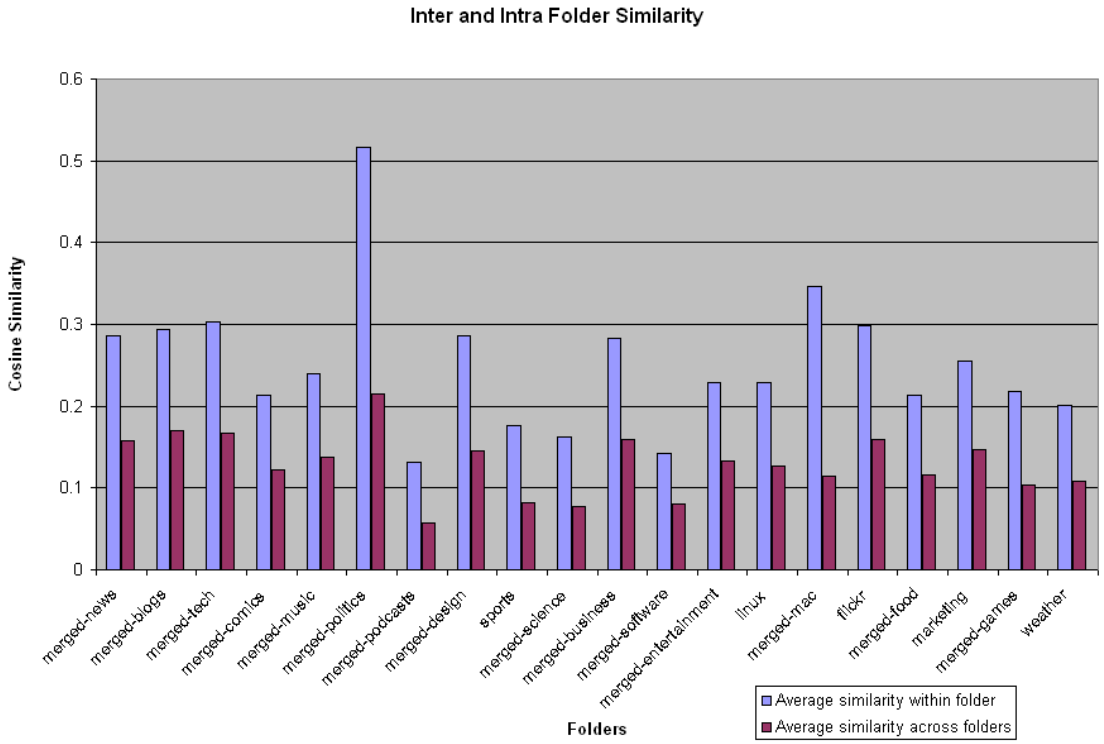


Figure 57: The average text similarity of the top 20 Folders. The chart shows the average similarity for the top 15 feeds within and across all the folders.

a topic, say photography, she could look at the tag cloud and quickly find feeds that are most often categorized under the folder name “photography”. Next, the system allows users to subscribe to the popular feeds directly in their Bloglines or Yahoo RSS readers. Alternatively, one could start a known feed and FTM! would provide recommendations based on the subscription information. “Feeds That Matter” has received a number of encouraging reviews especially from notable bloggers such as Micropersuation[10] and Lifehacker[11]. FTM! also has more than 500 bookmarks on delicious and our logs indicate that there is a steady stream of users who are actively using this service to find subscribe feeds in different categories.

5. Evaluation

We evaluate if folder similarity results in grouping related feeds together. We do this by comparing the folder similarity based on co-citations in URL vectors to text similarity of text obtained form the homepages of the feeds.

[10] http://www.micropersuation.com
[11] http://www.lifehacker.com

Figure 57 shows a comparison of average text similarity of feeds in the top 20 folders. For all the folders it was found that the feeds shared a greater similarity within the folder rather than across other folders. While the scores may seem low, studies on Technorati data by Brooks [21] show cosine similarity of posts sharing the same tag to be around 0.3. According to their study, when the same posts were clustered using the high scoring TFIDF terms the average text similarity was around 0.7.

Table 5. shows some of the recommendations for a few blogs. The feed recommendations are obtained by comparing the feeds to find how often they co-occur in the same folder. To evaluate the effectiveness of this system, we use the text based cosine similarity as a measure of how related the feeds are. We find that many of the recommended feeds have a high similarity score with the feed submitted. The best way to evaluate such a system would be through user studies and human evaluation of the results. We hope to perform such a study in the near future.

6. Conclusions

A number of Web applications and services can benefit from a set of intuitive, human understandable topic categories for feeds and blogs. This is especially true if we can also have a good 'training' set of feeds for each category. We found that public data from the Bloglines service provides the data from which to induce a set of topic categories and to associate a weighted set of feeds and blogs for each. We have presented a study of the Bloglines subscribers and have shown how folder names and subscriber counts can be used to find *feeds that matter* for a topic. We have also described it's use in applications such as feed recommendations and automatic identification of influential blogs. We have also implemented FTM!, a prototype site based on the algorithms described in this paper. This site has received encouraging reviews from the blogging community and positive feedback from active users.

http://www.dailykos.com	**Similarity**
http://www.andrewsullivan.com	**0.496**
http://www.talkingpointsmemo.com	**0.45**
http://atrios.blogspot.com	0.399
http://jameswolcott.com	**0.466**
http://mediamatters.org	0.262
http://yglesias.typepad.com/matthew/	0.285
http://billmon.org/	0.343
http://digbysblog.blogspot.com	**0.555**
http://instapundit.com/	0.397
http://www.washingtonmonthly.com/	**0.446**
http://blog.fastcompany.com	
http://business2.blogs.com/business2blog	0.303
http://www.fastcompany.com	0.454
http://sethgodin.typepad.com/seths_blog/	0.374
http://www.ducttapemarketing.com/	0.028
http://customerevangelists.typepad.com	0.399
http://blog.guykawasaki.com/	**0.441**
http://www.tompeters.com	**0.457**
http://www.paidcontent.org/	0.351
http://slashdot.org	
http://www.techdirt.com/	**0.516**
http://www.theregister.co.uk/	0.1
http://www.geeknewscentral.com/	0.286
http://www.theInquirer.net	0.2
http://news.com.com/	0.24
http://www.kuro5hin.org/	0.332
http://www.pbs.org/cringely/	0.087
http://backword.me.uk/	-
http://digg.com/	0.165
http://www.infoworld.com/news/index.html	0.203
http://www.yarnharlot.ca/blog/	
http://wendyknits.net/	**0.419**
http://www.woolflowers.net/	0.139
http://zeneedle.typepad.com/	0.383
http://www.keyboardbiologist.net/knitblog/	0.297
http://alison.knitsmiths.us/	0.284
http://knitandtonic.typepad.com/knitandtonic/	**0.542**
http://www.crazyauntpurl.com/	**0.521**
http://www.lollygirl.com/blog/	**0.4**
http://ma2ut.blogspot.com	**0.423**

Table V..5: Example recommendations for blogs in bold.

B. Epidemic Based Influence Models

1. Related Work

Research in the area of information propagation was inspired by a large body of work in disease and epidemic propagation. As described in [68] this model applies well in the Blogosphere where a blogger may have a certain level of interest in a topic and is thus *susceptible* to talking about it. By discussing the topic he/she may *infect* others and over time might *recover*. The authors use this approach in characterizing individuals into various phases of a topic in which they are more likely to become *infected*. They model individual propagation and use an expectation maximization algorithm to predict the likelihood of a blogger linking to another blogger. They also study the different types of topics present in the dataset and describe an approach to categorize topics into subtopics. Certain topics are more *infectious* than others and spread through the social network of bloggers. Automatically predicting such topics and developing models to accurately identify the propagation patterns on the Blogosphere is the main focus of this work.

Since bloggers are constantly keeping abreast of the latest news and often talk about new trends before they peak, recent research has focused on extracting opinions and identifying buzz from blogs [56]. Gruhl et al. [67] have found strong correlation between spikes in blog mentions to Amazon sales ranks of certain books. More recently, Lloyd et al. [130] found similar trends for named entities in blog mentions and RSS news feeds.

Blogs are often topical in nature and their link structures constantly evolve as new topics emerge. Ravi et al. [116] study the word burst models [106] and community structure on the Blogosphere [118]. They find a sustained and rapid increase in the size of the strongly connected component on the Blogosphere and explain that the community structure is due to the tendency of the bloggers to topically interlink with posts on other blogs.

2. Cascade Models

The link-based analysis in this section is based on the problem posed by [175] and influence models proposed by [103, 102]. These models aim to mathematically simulate the spread of information in social networks. Kempe et al. proposed an approximation of the NP-Hard problem of identifying a set of influential nodes to target so that we can maximize the number of nodes that are activated or influenced. We use the basic *Linear Threshold Model* as proposed by Kempe et al. While the original models were validated on citation graphs,

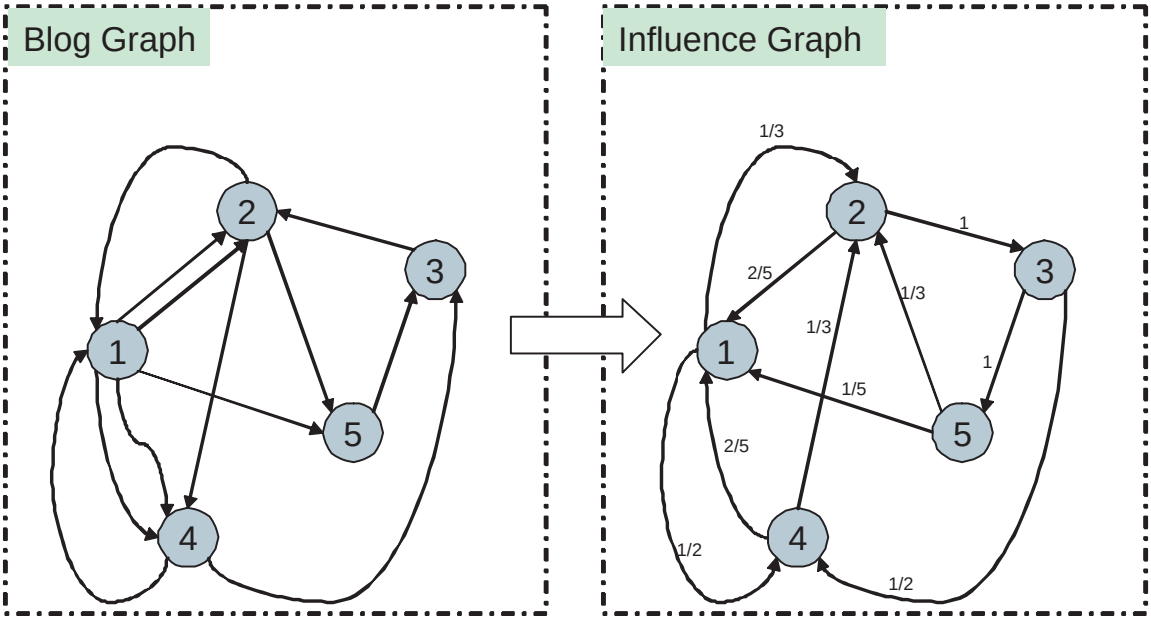


Figure 58: This diagram shows the conversion of a blog graph into an influence graph. A link from *u* to v indicates that *u* is influenced by *v*. The edges in the influence graph are reverse of the blog graph to indicate this influence. Multiple edges indicate stronger influence and are weighed higher

which are much smaller, we apply these algorithms on graphs derived from links between blogs. The citation graphs tend to be much cleaner and some of the techniques proposed do not apply well in the presence of splogs. We also discuss the applicability of simpler, PageRank-based heuristics for influence models on the Blogosphere and the web in general.

Bloggers generally tend to follow mainstream media, the Web, and also blog posts from people who may share similar interests. When an interesting *meme* emerges on some site, a blogger may choose to share it with his audience. Additionally, he may provide more insights and *trackback* to other sources of similar information. Other readers may comment on this post and thereby contribute to the conversation. Such an interactive process leads to the flow of information from one blogger to another. In approximating this interaction, we consider the presence of a link from site *u* to site *v* as evidence that the site *u* is *influenced by* site *v*. We consider only outlinks from posts and do not use comment links or trackback links for building the blog graphs. we take a rather simplistic view in the influence models and convert the *blog graph* to a directed *influence graph*. Figure 58 shows a hypothetical blog graph and its corresponding influence graph. An influence graph is a weighted, directed graph with edge weights indicating how much influence a particular source node has on its destination. Starting with the influence graph we aim to identify a set of nodes to target a piece of information such that it causes a large number of bloggers to be influenced by the idea.

Different influence models have been proposed [102, 103]. The two general categories are *Linear Threshold Model* and *Cascade Model*. We describe some of these below:

In the basic *Linear Threshold Model* each node has a certain threshold for adopting an idea or being influenced. The node becomes activated if the sum of the weights of the active neighbors exceeds this threshold. Thus if node *v* has threshold θ_v and edge weight b_{wv} such that neighbor *w* influenced *v*, then *v* becomes active only if

$$\sum_{w\ active\ neighbors\ of\ v} b_{wv} \geq \theta_v$$

and

$$\sum b_{wv} \leq 1$$

Another model is the *Independent Cascade Model* in which a node gets a single chance to activate each of its neighboring nodes and it succeeds with a probability P_{vw} which is independent of the history.

As described in the above model, we rank each directed edge between *u*, *v* in the *Influence Graph* such that the presence of multiple directed edges provides additional evidence that node *node u influences node v*. If $C_{u,v}$ is the number of parallel directed edges from *u* to *v* the edge weight

$$W_{u,v} = \frac{C_{u,v}}{d_v}$$

where d_v is the indegree of node *v* in the influence graph.

Since computing the optimal value of expected size of the influenced set, $\sigma(A)$, remains an open question, the algorithm runs the influence propagation model for pseudo-random threshold values and computes the approximate size of $\sigma(A)$.

In selecting the order of activation of nodes, the simplest ranking scheme is one using the number of inlinks (which corresponds to the outlinks in the influence graph). This represents how many other nodes can be influenced by activating the selected node. We also explored PageRank [165] as a heuristic in selecting the target set.

Finally we compare these heuristics with the greedy hill climbing algorithm. In the greedy approach nodes are incrementally added to the initial activation set without backtracking. At each time step, the influence model is run and a node is selected to be added to the initial target set. The node is selected such that adding it to the target set would provide the largest locally optimal increase in the size of the influenced node set.

Other methods such as “distance centrality” based heuristic are also widely used in many studies. This however could not be applied to the blog dataset since computing the centrality scores over large graphs is expensive without partitioning or identifying subgraphs.

3. Evaluation

The following section describes some of the experiments and results.

Weblog Dataset The dataset released by Intelliseek/Blogpulse[12] for the 2006 Weblogging Ecosystems Workshop consists of posts from about 1.3 million unique blogs. The data spans over 20 days during the time period in which there were terrorist attacks in London. This time frame witnessed significant activity in the Blogosphere with a number of posts pertaining to this subject. The link graph that we extracted from this dataset consists of 1.2 million links among 300K blogs. However it was also observed that Livejournal [13] sites tend to be highly interlinked and hence for the purpose of the influence models presented in the following sections, we do not consider blogs from these sites for inclusion in the initial activation set. However, we do not discard the blogs from the link graph.

In addition to the Blogpulse dataset we have used the publicly listed feed subscriptions from 82,428 users which consisted of 2,786,687 feeds in all, of which about 496,893 are unique. Bloglines allows users to organize their subscriptions in folders. Although only 35% of Bloglines subscribers use this feature, it provides substantial data to categorize the feeds into different topics.

We run the influence models using different heuristics such as PageRank, indegre and greedy algorithm. In PageRank and indegree the nodes are added to the initial target set in the order of their rank. Once a node is selected, the influence model is run and nodes are activated depending on their threshold. The number of nodes influenced is estimated over multiple iterations. In greedy algorithm nodes are incrementally added to the target set if they locally maximize the size fo the influenced node set.

We first eliminate spam blogs from the collection using the algorithm previously described. As seen from results in the 59, after eliminating splogs, the top results obtained from the indegree heuristics almost approximated PageRank. This was also due to the fact that about 70% of the blogs as ranked by PageRank and indegree match after splog elimination. However, it was found that the PageRank and greedy heuristics seem to perform almost the same even after the elimination of roughly 103687 nodes which correspond to

[12] http://www.blogpulse.com
[13] http://livejournal.com

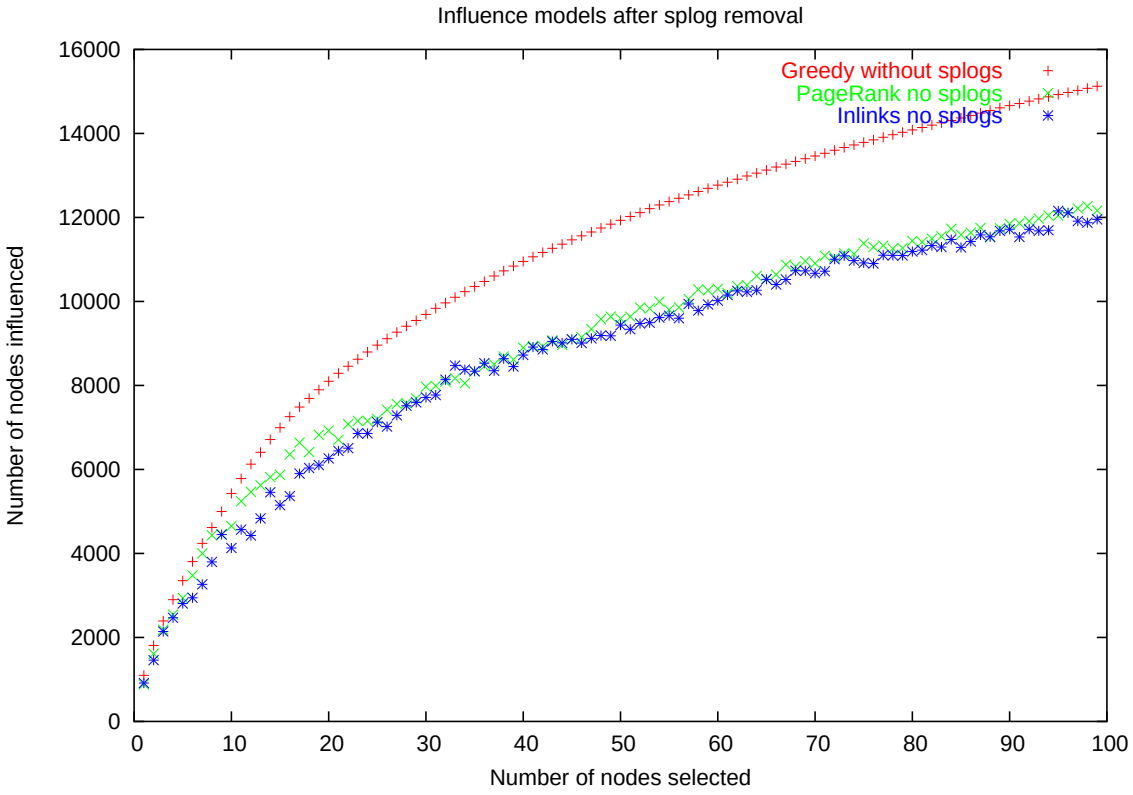


Figure 59: *The graph shows the performance of the indegree and pagerank heuristic after splog elimination. The above results show the average influence after 10 iterations of each heuristic.*

splogs (including failed URLs).

The Greedy heuristic of node selection performs better than both PageRank or indegree. However one of the disadvantages of the greedy approach is that it is computationally quite expensive. PageRank on the other hand is an iterative algorithm that converges to the principal eigenvector of the adjacency matrix. While it is faster to compute, it requires knowledge of the structure of links which might emerge only after the Blogpost has been read and linked to by other blogs over a period of time.

4. Identifying Leaders using Influence Propagation

Consider a scenario where a user has a few blogs that she subscribes to or is familiar with a couple of extremely popular blogs for a topic. Now, she wishes to find other blogs that are also opinion leaders in this area. In this section, we present a simple method that is based on an influence propagation model using linear threshold. In the following section, we use the blog graph data from the 2006 Workshop on Weblogging Ecosystems (WWE). This dataset consists of posts from about 1.3M blogs and spans over a

period of 20 days. Using a few authoritative blogs obtained from the Bloglines data, the technique identifies other topically authoritative blogs.

To propagate influence, starting from the seed set, we use the basic *Linear Threshold Model* [102, 103] in which each node has a certain threshold for adopting an idea or being influenced. The node becomes activated if the sum of the weights of the active neighbors exceeds this threshold. Thus if node *v* has threshold θ_v and edge weight b_{wv} such that neighbor *w* influenced *v*, then *v* becomes active only if

$$\sum_{w\ active\ neighbors\ of\ v} b_{wv} \geq \theta_v$$

and

$$\sum b_{wv} \leq 1$$

As described in Java et al. [93], we consider the presence of a link from site *u* to site *v* as evidence that the site *u* is *influenced by* site *v*. Using the above model, we rank each directed edge between *u*, *v* in the *Influence Graph* such that the presence of multiple directed edges provides additional evidence that node *node u influences node v*. If $C_{u,v}$ is the number of parallel directed edges from *u* to *v* the edge weight

$$b_{v,w} = \frac{C_{v,w}}{d_w}$$

where d_v is the indegree of node *v* in the influence graph.

The Identifying Leaders Using Influence Propagation (ILIP) algorithm described in Algorithm 3 finds a set of nodes that are influential for a given topic. As shown in figure 60, we start with some seed blogs for a given topic and induce a set of blogs that are termed as the *followers.* Followers are those blogs that are often influenced by the seed set. The goal is to infer other authoritative blogs or *leaders* for the topic. By iterating the linear threshold influence propagation model over the entire blog graph, we can find other blogs that are topically similar to the seed set and are also authoritative. The pseudocode of the ILIP Algorithm 3 describes the various steps involved in identifying topical influential nodes. Starting with a few top ranked feeds from the Bloglines dataset for the folders ‘Politics’, ‘Tech’,‘Business’ and ‘Knitting’ we use the ILIP algorithm to find other leaders in the blog graph. Table V..6 to V..8 show some of the results.

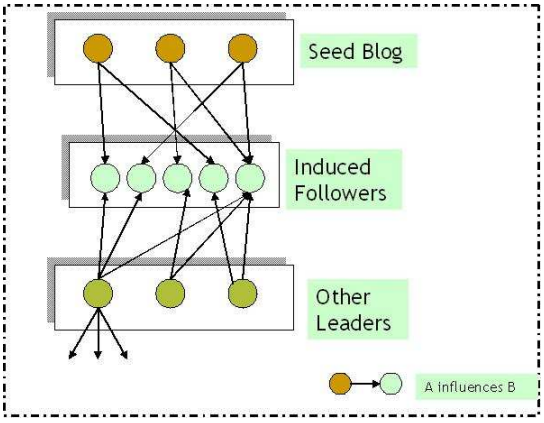


Figure 60: Identifying Leaders Using Information Propagation (ILIP): Starting with a few seed blogs on a topic a set of followers are induced and other leaders for this set are identified using the influence graph.

Algorithm 3 Identifying Leaders Using Influence Propagation (ILIP) Algorithm

$S \leftarrow SeedSet$
$F \leftarrow InfluencedFollowersSet$
$IG \leftarrow InfluenceGraph$
for all i such that $0 \leq i \leq max\_iterations$ **do**
  Activate S
  for all $v \in IG$ **do**
    θ_v = random score
  end for
  for all $v \in IG$ **do**
    if $\sum_{w\ active\ neighbors\ of\ v} b_{wv} \geq \theta_v$ **then**
      Activate v
      add v to F_i
    end if
  end for
end for
$F = F_i \bigcup F_{i+1} \bigcup \cdots F_{max\_iterations}$
for all k has inlinks to F **do**
  o_k = outlink count of k
  n_k = number of nodes linked from k to F
  $leader\_score = \frac{n_k}{o_k} * log(o_k)$
end for

Seed Blogs
http://www.dailykos.com
http://www.talkingpointsmemo.com
Top Leader Blogs
http://www.huffingtonpost.com/theblog
http://americablog.blogspot.com
http://thinkprogress.org
http://www.tpmcafe.com
http://www.crooksandliars.com
http://atrios.blogspot.com
http://www.washingtonmonthly.com
http://billmon.org
http://www.juancole.com
http://capitolbuzz.blogspot.com
http://instapundit.com
http://www.opinionjournal.com
http://digbysblog.blogspot.com
http://michellemalkin.com
http://www.powerlineblog.com
http://theleftcoaster.com
http://www.andrewsullivan.com
http://www.thismodernworld.com

Table V..6: Leaders found using ILIP for topic **'Politics'**

Seed Blogs
http://slashdot.org
http://www.kuro5hin.org
Top Leader Blogs
http://www.boingboing.net
http://www.engadget.com
http://www.metafilter.com
http://www.c10n.info
http://www.makezine.com/blog
http://radio.weblogs.com/0001011
http://mnm.uib.es/gallir
http://www.mozillazine.org
http://weblogs.mozillazine.org/asa
http://www.gizmodo.com

Table V..7: Leaders found using ILIP for topic **'Technology'**

Seed Blogs
http://www.yarnharlot.ca/blog
http://wendyknits.net
Top Leader Blogs
http://booshay.blogspot.com
http://mamacate.typepad.com/mamacate
http://www.thejonblog.com/knit
http://alison.knitsmiths.us
http://www.dioramarama.com/kmel
http://knittersofdoom.blogspirit.com
http://tonigirl.blogdrive.com
http://www.crazyauntpurl.com
http://www.januaryone.com
http://nathaniaapple.typepad.com/knit_quilt_stitch
http://www.knittygritty.net
http://www.katwithak.com
http://www.myblog.de/evelynsbreiwerk
http://nepenthe.blog-city.com
http://zardra.blogspot.com

Table V..8: The Leaders found using ILIP for topic **'Knitting'**

5. Conclusions

We have presented how simple epidemic propagation based influence models can be enhanced by using a few examples or seed nodes from which the propagation starts. The Identifying Leaders Using Influence Propagation (ILIP) algorithm works by propagating the influence from the seed nodes to it's neighbors in iterative steps, akin to a random walk process. We find that this mechanism leads to the discovery of a topically set of blogs that follow the seed sets, but also that it is capable of finding other related, influential blogs based on the co-citation information from the follower set. This techneque has applications in influence prediction, recommendation and viral marketing.

Chapter VI.

CONCLUSIONS

In this dissertation, we have presented a framework for analyzing social media content and structure. Social media tools and user generated content is changing how we find, access and share information. At the same time the easy availability of large, social data makes it possible for computer scientists and social scientists to pose research questions on the scale that was never possible earlier. This however, requires us to rethink the approaches to algorithmic analysis of Web data. Social computing requires developing a broad range of tools and paradigms that are more applicable for working with such data. The work presented in this dissertation focuses on two key aspects: analyzing social media content and analyzing social media structure.

In the thesis statement we stated that an effective understanding and modeling of interactions and communication behavior in social media is enabled through the combined analysis of its special properties, structure and content. The direct contribution of this work is in the development of new tools for analyzing social media content and detecting communities in social graphs. In doing so our algorithms and techniques take advantage of the special properties of social media.

First, we explored extracting semantic information from RSS streams and news feeds. The goal was to create a structured representation of unstructured text. Although understanding natural language text is a difficult problem, we find that in a restricted domain (such as news stories), it is possible to effectively construct the text meaning representation of the news summary. The main contribution of this work is to demonstrate the feasibility of extracting knowledge from free text and constructing semantic fact repositories that can be used to perform structured queries over natural language text.

Next, we describe how opinions and sentiments are identified from social media content. We describe BlogVox, a system built to index, retrieve and rank opinions on a given query term. In developing the BlogVox

system, we describe some of the challenges in processing large scale blog corpora. We developed novel techniques for dealing with spam and focusing on the content of the blog post (as opposed to the navigation and advertising content). We evaluate the effectiveness of our methods in the settings of the TREC blog track. The significant contribution of this work is the development of data cleaning methods in social media and multiple scoring mechanisms for ranking opinions on the blogosphere.

The second section of this dissertation concentrates on the community extraction component of the thesis statement. Communities are what makes social media exciting and identifying it's structure and content is a significant challenge. Here, we highlight the importance of using the additional meta-data that tags and folksonomies provide in social media. The tags are an additional way to constrain the grouping of related blogs. Thus we extend the definition of a community and describe it as a set of nodes that are more closely linked to each other than the rest of the network and share a common set of descriptive tags. We show that this extended definition of a community is useful particularly when working with social graphs and yields improved community structures.

Extracting communities from large graphs is computationally expensive and many algorithms do not scale well to the size of social media datasets. We describe a novel algorithm to detect communities based on a sampling approach. Many social graphs have a power law distribution and it is possible to identify the broad community structure of the network by analyzing the link structure of the dense, core region of the graph and approximating the membership of the remaining nodes by analyzing the links into the core region. This surprisingly simple, yet intuitive approach to community detection performs well and has a low computational and memory overhead.

Techniques developed in this thesis has been applied to several real world datasets. In particular, we explore the community structure and content of microblogging. Ours is the first study in the literature that provides an in-depth analysis of microblogging phenomenon and describes a taxonomy of user intentions in such environments. The BlogVox system and the influence models described in this thesis made use of the TREC blog collection, which is a large dataset of over 3.2 Million posts. We address several challenges in dealing with such large scale datasets. Finally, we describe clustering algorithms for mining blog feed subscriptions across of over 83K users. The results of our analysis have provided deeper insights into the feed subscription patterns and usage. Moreover, we show how subscription information can be applied to the feed distillation task and provides an intuitive way to group blogs and identify the "*Feeds That Matter*" for various topics.

A. Discussion

The broader impact of this work is to understand online, human communications and study how various elements of social media tools and platforms facilitate this goal. Our study spans a period of three years and is a snapshot into the World Wide Web's changing landscape. We see the emergence of social media and it's mainstream adoption as a key factor that has brought about a substantial change in how we interact with each other.

Through this study we have found that blogs are an important component of social media. The ease of publishing and ability to freely express thoughts, opinions and comments provided by blogs is unprecedented. The study of opinions as expressed through blogs gives us a view into the collective minds of a demographic, geographic region or even a generation. It's impacts have been felt on the success or failure of products, political campaigns as well as social change. While understanding natural language text remains the ultimate AI challenge, we are slowly getting closer to it every day. Bits of information, sentiments and the meaning expressed in the text can be gleaned by both semantic and syntactic processing.

Communities are the basis of organization in human society and we see a reflection of our offline associations in our online interactions. The social web infrastructure is built on facilitating interactions between individuals who share similar interests. Communities emerge through our shared actions (like use of similar tags, rating videos or subscribing to feeds), by explicit linking (via blogs, adding friends to the social graphs and through trackbacks from comments) and by implicit behavior (like expressing interest in a topic, clicking through results from a search engine or clicking online ads). A holistic approach to community detection needs to consider the multiple dimensions of our online presence. While this study takes an initial step in this direction, it presents novel algorithms that make use of social context like tags or the long tail distribution of links and attention on the Web. These techniques presented here enhance the existing approaches to community extraction by integrating the meta-data, which is useful in describing communities.

Throughout our study of social media datasets, our goals has been to understand social behavior. The advent of microblogging provided an interesting opportunity to study this new trend itself, and the user intentions in this specific setting. The study of Twitter, a microblogging environment through the collective analysis of the content of updates made and the social network revealed several interesting properties. We find that people use microblogging to talk about their daily activities and to seek or share information. Studying the user intentions associated at a community level show how users with similar intentions connect with each other.

As the number of blogs online are increasing, it is becoming increasingly difficult to find relevant feeds to subscribe to. In an effort to address this problem, we describe a readership-based approach to group related feeds and rank feeds for different topics. While this was the original motivation for our work, it revealed some fascinating cues into the subscription patterns of a large collection of users. For example, about 35% of users organize their feeds in folders and although users might have a varied set of descriptive terms for a feed, a common consensus emerges. A large fraction of users subscribe to less than 30 feeds and as the number of feeds subscribed increase, there is a greater need for organization and more folders are used to categorize them. We also describe how such a system can be applied for feed recommendation task where a user can be recommended new feed to subscribe based on her current subscriptions.

B. Future Work and Open Problems

The growth of social media sites and it's varied applications is engaging a wider range of audience and demographics. A significant share of our attention is attracted by social networking sites, photo and video sharing applications and social bookmarks sites, among many others.

In this study we have analyzed a shard of our online activities on these social media sites. It also leads us to some challenging new questions and open problems. Broadly, these can be identified under four main categories:

- **Content Analysis** Open domain, natural language processing of free-form, unstructured, text remains challenging. Content on the blogosphere adds to the layer of complexity. While some problems and data cleaning approaches are addressed in this work, there are a number of challenges in dealing with blog datasets. For example, due to the informal nature of text in blogs, slangs, neologisms and other constructs are quite common. However, most NLP tools find it difficult to process these types of inputs. This impacts the performance of opinion retrieval and other tasks.

- **Temporal Analysis** The algorithms presented in this thesis mainly analyze static graphs. However, most social datasets have a temporal aspect to them. Mining evolving, temporal graph data for community structure, trend detection, topic evolution and concept drift is a new field of research that has many interesting applications.

- **Community Analysis** Our approach takes a simplistic view of a community. In this definition, a community is a set of nodes that have more links to each other than the rest of the network. However, in the

real world there can be many ways to define what constitutes a community. For example, users who view similar videos on YouTube might form a community and people belonging to the same geographical area might be another community, etc. Many community detection algorithms work only in one dimension (usually, relying on link information alone) and membership is often exclusive. Discovering partial membership and multi-dimensional communities is a challenging problem and something worth investigating further.

- **Blog Search Implications** When searching for information on a blog search engine users are often looking for relevant information on recent events. Results may be ranked by a factor of recency, relevance and authority. It is an open question as to what are the parameters to combine these factors effectively. Further, queries to blog search engines are made to either search for content in the post or for finding relevant feeds. This requires research into two different strategies for ranking blog information. Another interesting question that arises with respect to blog search is index quality and freshness. It is important that blog search infrastructure is capable of quickly indexing new posts. There is a tradeoff between index quality and freshness and discovering good strategies for indexing blog content is of critical importance and an open research problem. Lastly, as social media content becomes even more pervasive, we find that Web search engine queries also return a number of blog posts within their results. It is an open question as to how this effects Web search ranking.

BIBLIOGRAPHY

[1] Institute for language and information technologies. `http://ilit.umbc.edu/`.

[2] *Efficient Identification of Web Communities*, 2000.

[3] Xml schema part 0: Primer. World Wide Web Consortium Specification, 2004. see `http://www.w3.org/TR/xmlschema-0/`.

[4] Lada A. Adamic, Bernardo A. Huberman;, A. Barab'asi, R. Albert, H. Jeong, and G. Bianconi;. Power-law distribution of the world wide web. *Science*, 287(5461):2115a+, March 2000.

[5] Reka Albert and Albert-Laszlo Barabasi. Statistical mechanics of complex networks, Jun 2001.

[6] J.F. Allen. Recognizing intentions from natural language utterances. *Computational Models of Discourse*, pages 107–166, 1983.

[7] J.L. Austin. *How to Do Things with Words*. Oxford University Press Oxford, 1976.

[8] Lars Backstrom, Dan Huttenlocher, Jon Kleinberg, and Xiangyang Lan. Group formation in large social networks: membership, growth, and evolution. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 44–54, New York, NY, USA, 2006. ACM Press.

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

[10] B. Lavoie-M. McShane S. Nirenburg Beale, S. and T. Korelsky. Question answering using ontological semantics.

[11] S. Beale, S. Nirenburg, and K. Mahesh. Semantic analysis in the mikrokosmos machine translation project, 1995.

[12] P.J. Beltran-Ferruz, P.A. ez Caler, and P.Gervas. Converting frames into OWL: Preparing Mikrokosmos for linguistic creativity. In *LREC Workshop on Language Resources for Linguistic Creativity*, 2004.

[13] P.J. Beltran-Ferruz, P.A. Gonzalez-Caler, and P.Gervas. Converting Mikrokosmos frames into description logics. In *RDF/RDFS and OWL in Language Technology: 4th ACL Workshop on NLP and XML*, July 2004.

[14] Indrajit Bhattacharya and Lise Getoor. Relational clustering for multi-type entity resolution. In *The ACM SIGKDD Workshop on Multi Relational Data Mining (MRDM)*, Chicago, IL, USA, 2005.

[15] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, 2003.

[16] Blogpulse. The 3rd annual workshop on weblogging ecosystem: Aggregation, analysis and dynamics, 15th world wide web conference, May 2006.

[17] Stephen P. Borgatti and Martin G. Everett. Models of core/periphery structures. *Social Networks*, 21(4):375–395, October 2000.

[18] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the web. *Computer Networks*, 33(1):309–320, June 2000.

[19] Andrei Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, 2002.

[20] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. Graph structure in the web. In *Proceedings of the 9th international World Wide Web conference on Computer networks : the international journal of computer and telecommunications netowrking*, pages 309–320, Amsterdam, The Netherlands, The Netherlands, 2000. North-Holland Publishing Co.

[21] Christopher H. Brooks and Nancy Montanez. Improved annotation of the blogosphere via autotagging and hierarchical clustering. In *WWW*, 2006.

[22] Ciro Cattuto, Vittorio Loreto, and Luciano Pietronero. Collaborative tagging and semiotic dynamics. *CoRR*, abs/cs/0605015, 2006.

[23] Ciro Cattuto, Vittorio Loreto, and Luciano Pietronero. Collaborative tagging and semiotic dynamics, 2006.

[24] Soumen Chakrabarti, Byron Dom, Prabhakar Raghavan, Sridhar Rajagopalan, David Gibson, and Jon M. Kleinberg. Automatic resource compilation by analyzing hyperlink structure and associated text. *Computer Networks*, 30(1-7):65–74, 1998.

[25] Remy Fiorentino Sarah Glass Charlene Li, Josh Bernoff. Social technographics mapping participation in activities forms the foundation of a social strategy, 2007.

[26] Ed H. Chi and Todd Mytkowicz. Understanding the efficiency of social tagging systems using information theory. In *HT '08: Proceedings of the nineteenth ACM conference on Hypertext and hypermedia*, pages 81–88. ACM, 2008.

[27] Yun Chi, Xiaodan Song, Dengyong Zhou, Koji Hino, and Belle L. Tseng. Evolutionary spectral clustering by incorporating temporal smoothness. In *KDD*, pages 153–162, 2007.

[28] Yun Chi, Shenghuo Zhu, Xiaodan Song, Jun'ichi Tatemura, and Belle L. Tseng. Structural and temporal analysis of the blogosphere through community factorization. In *KDD*, pages 163–172, 2007.

[29] Fan R. K. Chung. *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92) (Cbms Regional Conference Series in Mathematics)*. American Mathematical Society, February 1997.

[30] Aaron Clauset, M. E. J. Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical Review E*, 70:066111, 2004.

[31] Aaron Clauset, Cosma R. Shalizi, and M. E. J. Newman. Power-law distributions in empirical data, Jun 2007.

[32] Comscore. Sites for social butterflies. `http://www.usatoday.com/tech/webguide/2007-05-28-social-sites_N.htm`, May 2007.

[33] R. Scott Cost, Tim Finin, Anupam Joshi, Yun Peng, Charles Nicholas, Ian Soboroff, Harry Chen, Lalana Kagal, Filip Perich, Youyong Zou, and Sovrin Tolia. ITtalks: A Case Study in the Semantic Web and DAML+OIL. *IEEE Intelligent Systems Special Issue*, January 2002.

[34] J.A. Costa and III Hero, A.O. Classification constrained dimensionality reduction. *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*, 5:v/1077–v/1080 Vol. 5, 2005.

[35] H. Akkermans et al. D. Fensel, F. van-Harmelen.

[36] Olivier Dameron, Daniel L. Rubin, and Mark A. Musen. Challenges in converting frame-based ontology into OWL: the Foundational Model of Anatomy case-study. In *American Medical Informatics Association Conference AMIA05*, 2005.

[37] Kushal Dave, Steve Lawrence, and David M. Pennock. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *WWW*, pages 519–528, 2003.

[38] Ravi Kumar Prabhakar Raghavan David Liben-Nowell, Jasmine Novak and Andrew Tomkins. Geographic routing in social networks. *Proceedings of the National Academy of Sciences*,, 102(33):11623–1162, 2005.

[39] Drew McDermott Dejing Dou and Peishen Qi. Ontology translation by ontology merging and automated reasoning. In *Proc. EKAW Workshop on Ontologies for Multi-Agent Systems*, 2002.

[40] Imre Derenyi, Gergely Palla, and Tamas Vicsek. Clique percolation in random networks. *Physical Review Letters*, 94:160202, 2005.

[41] Inderjit S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 269–274, New York, NY, USA, 2001. ACM.

[42] S. Dill, N. Eiron, D. Gibson, D. Gruhl, and R. Guha. Semtag and seeker: Bootstrapping the semantic web via automated semantic annotation. In *Proceedings of WWW2003*, 2003.

[43] Li Ding, Tim Finin, Anupam Joshi, Rong Pan, R. Scott Cost, Yun Peng, Pavan Reddivari, Vishal C Doshi, , and Joel Sachs. Swoogle: A search and metadata engine for the semantic web. In *Proceedings of the Thirteenth ACM Conference on Information and Knowledge Management*, 2004.

[44] D. Donato, L. Laura, S. Leonardi, and S. Millozzi. Large scale properties of the webgraph. *European Physical Journal B*, 38:239–243, March 2004.

[45] Drineas, Frieze, Kannan, Vempala, and Vinay. Clustering in large graphs and matrices. In *SODA: ACM-SIAM Symposium on Discrete Algorithms (A Conference on Theoretical and Experimental Analysis of Discrete Algorithms)*, 1999.

[46] Micah Dubinko, Ravi Kumar, Joseph Magnani, Jasmine Novak, Prabhakar Raghavan, and Andrew Tomkins. Visualizing tags over time. In *WWW*, 2006.

[47] J. Duch and A. Arenas. Community detection in complex networks using extremal optimization. *Physical Review E*, 72:027104, 2005.

[48] J. Elsas, J. Arguello, J. Callan, and J. Carbonell. Retrieval and feedback models for blog distillation. In *The Sixteenth Text REtrieval Conference (TREC 2007) Proceedings*, 2007.

[49] Jesse English. Dekade ii: An environment for development and demonstration in natural language processing, 2006.

[50] Gary William Flake, Steve Lawrence, C. Lee Giles, and Frans Coetzee. Self-organization of the web and identification of communities. *IEEE Computer*, 35(3):66–71, 2002.

[51] Charless Fowlkes, Serge Belongie, Fan Chung, and Jitendra Malik. Spectral grouping using the nyström method. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(2):214–225, 2004.

[52] Natalie Glance Gilad Mishne. Predicting movie sales from blogger sentiment. In *AAAI 2006 Spring Symposium on Computational Approaches to Analysing Weblogs (AAAI-CAAW 2006)*, 2006.

[53] Daniel Gildea and Daniel Jurafsky. Automatic labeling of semantic roles. In *Computational Linguistics*, pages 245–288.

[54] Michelle Girvan and M. E. J. Newman. Community structure in social and biological networks, Dec 2001.

[55] Michelle Girvan and M. E. J. Newman. Community structure in social and biological networks, Dec 2001.

[56] Natalie Glance, Matthew Hurst, Kamal Nigam, Matthew Siegler, Robert Stockton, and Takashi Tomokiyo. Deriving marketing intelligence from online discussion. In *KDD*, pages 419–428, 2005.

[57] Natalie S. Glance, Matthew Hurst, Kamal Nigam, Matthew Siegler, Robert Stockton, and Takashi Tomokiyo. Deriving marketing intelligence from online discussion. In *KDD*, pages 419–428, 2005.

[58] Eric Glover, David M. Pennock, Steve Lawrence, and Robert Krovetz. Inferring hierarchical descriptions. In *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, pages 507–514, New York, NY, USA, 2002. ACM Press.

[59] Scott Golder and Bernardo A. Huberman. Usage patterns of collaborative tagging systems. *Journal of Information Science*, 32(2):198–208, April 2006.

[60] Sam Gosling. *Snoop: What Your Stuff Says About You*. Basic Books, 2008.

[61] Mark S. Granovetter. The strength of weak ties. *The American Journal of Sociology*, 78(6):1360–1380, 1973.

[62] H.P. Grice. Utterers meaning and intentions. *Philosophical Review*, 78(2):147–177, 1969.

[63] Ralph Grishman and Beth Sundheim. Message understanding conference-6: a brief history. In *Proceedings of the 16th Conference on Computational Linguistics*, pages 466–471, 1996.

[64] William G. Griswold. Five enablers for mobile 2.0. *Computer*, 40(10):96–98, October 2007.

[65] Robert Grone, Russell Merris, and V. S. Sunder. The laplacian spectrum of a graph. *SIAM J. Matrix Anal. Appl.*, 11(2):218–238, 1990.

[66] Barbara J. Grosz. *Focusing and Description in Natural Language Dialogues*. Cambridge University Press, New York, New York, 1981.

[67] Daniel Gruhl, R. Guha, Ravi Kumar, Jasmine Novak, and Andrew Tomkins. The predictive power of online chatter. In *KDD*, pages 78–87, 2005.

[68] Daniel Gruhl, R. Guha, David Liben-Nowell, and Andrew Tomkins. Information diffusion through blogspace. In *WWW*, pages 491–501, 2004.

[69] Mika Gustafsson, Michael Hrnquist, and Anna Lombardi. Comparison and validation of community structures in complex networks. In *Physica A: Statistical Mechanics and its Application,.367*, pages 559–576, 2006.

[70] Marike Guy and Emma Tonkin. Folksonomies: Tidying up tags. *D-Lib Magazine*, 12(1), 2006.

[71] Zoltán Gyöngyi and Hector Garcia-Molina. Web spam taxonomy. In *First International Workshop on Adversarial Information Retrieval on the Web*, 2005.

[72] Zoltán Gyöngyi, Hector Garcia-Molina, and Jan Pedersen. Combating web spam with TrustRank. In *Proceedings of the 30th International Conference on Very Large Databases*, pages 576–587. Morgan Kaufmann, 2004.

[73] J. Ham, D. Lee, S. Mika, and B. Scholkopf. A kernel view of the dimensionality reduction of manifolds, 2004.

[74] Susan C. Herring, Inna Kouper, John C. Paolillo, Lois Ann Scheidt, Michael Tyworth, Peter Welsch, Elijah Wright, and Ning Yu. Conversations in the blogosphere: An analysis "from the bottom up". In *HICSS '05: Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 4*, page 107.2, Washington, DC, USA, 2005. IEEE Computer Society.

[75] Paul Heymann and Hector Garcia-Molina. Collaborative creation of communal hierarchical taxonomies in social tagging systems. Technical Report 2006-10, Stanford University, April 2006.

[76] Paul Heymann, Georgia Koutrika, and Hector Garcia-Molina. Can social bookmarking improve web search? In *WSDM '08: Proceedings of the international conference on Web search and web data mining*, pages 195–206, New York, NY, USA, 2008. ACM.

[77] Paul Heymann, Daniel Ramage, and Hector Garcia-Molina. Social tag prediction. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 531–538, New York, NY, USA, 2008. ACM.

[78] Andrew Hogue and David R. Karger. Thresher: Automating the unwrapping of semantic content from the world wide web. In *Proceedings of the Fourteenth International World Wide Web Conference*, May 2005.

[79] Petter Holme. Core-periphery organization of complex networks. *Physical Review E*, 72:046111, 2005.

[80] Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, and Mike Dean. Swrl: A semantic web rule language combining owl and ruleml. World Wide Web Consortium Specification, May 2004.

[81] Ian Horrocks, Peter F. Patel-Schneider, and Frank van Harmelen. From shiq and rdf to owl: the making of a web ontology language. *J. Web Sem.*, 1(1):7–26, 2003.

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

[83] C. Schnorr J. Keuchel. Efficient graph cuts for unsupervised image segmentation. using probabilistic sampling and svd-based approximation, 2003.

[84] Akshay Java. Global distribution of twitter users. `http://ebiquity.umbc.edu/blogger/2007/04/15/global-distribution-of-twitter-users/`.

[85] Akshay Java, Tim Finin, and Sergei Nirenburg. Integrating language understanding agents into the semantic web. In Terry Payne and Valentina Tamma, editors, *Proceedings of the AAAI Fall Symposium on Agents and the Semantic Web*, November 2005.

[86] Akshay Java, Tim Finin, and Sergei Nirenburg. Text understanding agents and the Semantic Web. In *Proceedings of the 39th Hawaii International Conference on System Sciences*, Kauai HI, January 2006.

[87] Akshay Java, Anupam Joshi, and Tim Finin. Approximating the community structure of the long tail. In *Submitted to the Second International Conference on Weblogs and Social Media*, November 2007.

[88] Akshay Java, Pranam Kolari, Tim Finin, Anupam Joshi, and Tim Oates. Feeds That Matter: A Study of Bloglines Subscriptions. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM 2007)*. Computer Science and Electrical Engineering, University of Maryland, Baltimore County, March 2007.

[89] Akshay Java, Pranam Kolari, Tim Finin, Anupam Joshi, and Tim Oates. Feeds That Matter: A Study of Bloglines Subscriptions. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM 2007)*, 2007.

[90] Akshay Java, Pranam Kolari, Tim Finin, James Mayfield, Anupam Joshi, and Justin Martineau. The UMBC/JHU blogvox system. In *Proceedings of the Fifteenth Text Retrieval Conference*, November 2006.

[91] Akshay Java, Pranam Kolari, Tim Finin, James Mayfield, Anupam Joshi, and Justin Martineau. BlogVox: Separating Blog Wheat from Blog Chaff. In *Proceedings of the Workshop on Analytics for*

Noisy Unstructured Text Data, 20th International Joint Conference on Artificial Intelligence (IJCAI-2007), 2007.

[92] Akshay Java, Pranam Kolari, Tim Finin, and Tim Oates. Modeling the Spread of Influence on the Blogosphere. Technical report, University of Maryland, Baltimore County, March 2006.

[93] Akshay Java, Pranam Kolari, Tim Finin, and Tim Oates. Modeling the Spread of Influence on the Blogosphere. Technical report, University of Maryland, Baltimore County, March 2006.

[94] Akshay Java, Xiaodan Song, Tim Finin, and Belle Tseng. Why we twitter: understanding microblogging usage and communities. In *WebKDD/SNA-KDD '07: Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pages 56–65, New York, NY, USA, 2007. ACM.

[95] Rie Johnson and Tong Zhang. Graph-based semi-supervised learning and spectral kernel design. In *IEEE Trans. Info. Theory*, 2007.

[96] Anubhav Kale, Amit Karandikar, Pranam Kolari, Akshay Java, Anupam Joshi, and Tim Finin. Modeling Trust and Influence in the Blogosphere Using Link Polarity. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM 2007)*, March 2007.

[97] In-Ho Kang and GilChang Kim. Query type classification for web document retrieval. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 64–71, New York, NY, USA, 2003. ACM Press.

[98] Amit Karandikar, Akshay Java, Anupam Joshi, Tim Finin, Yelena Yesha, and Yaacov Yesha. Second Space: A Generative Model For The Blogosphere. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM 2008)*. AAAI, AAAI Press, March 2008.

[99] Brian Karrer, Elizaveta Levina, and M. E. J. Newman. Robustness of community structure in networks, Sep 2007.

[100] Brian Karrer, Elizaveta Levina, and M. E. J. Newman. Robustness of community structure in networks, Sep 2007.

[101] George Karypis and Vipin Kumar. *MeTis: Unstrctured Graph Partitioning and Sparse Matrix Ordering System, Version 2.0*, 1995.

[102] David Kempe, Jon M. Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *KDD*, pages 137–146, 2003.

[103] David Kempe, Jon M. Kleinberg, and Éva Tardos. Influential nodes in a diffusion model for social networks. In *ICALP*, pages 1127–1138, 2005.

[104] P. Kingsbury, M. Palmer, and M. Marcus. Adding semantic annotation to the penn treebank. In *Proceedings of the Human Language Technology Conference (HLT'02)*, 2002.

[105] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[106] Jon M. Kleinberg. Bursty and hierarchical structure in streams. *Data Min. Knowl. Discov.*, 7(4):373–397, 2003.

[107] Pranam Kolari. Welcome to the splogosphere: 75% of new pings are spings(splogs), 2005. [Online; accessed 22-December-2005; `http://ebiquity.umbc.edu/blogger/?p=429`].

[108] Pranam Kolari, Tim Finin, and Anupam Joshi. SVMs for the Blogosphere: Blog Identification and Splog Detection. In *Proceedings of the AAAI Spring Symposium on Computational Approaches to Analysing Weblogs*. AAAI Press, March 2006.

[109] Pranam Kolari, Tim Finin, Yelena Yesha, Yaacov Yesha, Kelly Lyons, Stephen Perelgut, and Jen Hawkins. On the Structure, Properties and Utility of Internal Corporate Blogs. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM 2007)*, March 2007.

[110] Pranam Kolari, Akshay Java, and Tim Finin. Characterizing the Splogosphere. In *Proceedings of the 3rd Annual Workshop on Weblogging Ecosystem: Aggregation, Analysis and Dynamics, 15th World Wid Web Conference*. Computer Science and Electrical Engineering, May 2006.

[111] Pranam Kolari, Akshay Java, and Tim Finin. Characterizing the Splogosphere. In *Proceedings of the 3rd Annual Workshop on Weblogging Ecosystem: Aggregation, Analysis and Dynamics, 15th World Wid Web Conference*, May 2006.

[112] Pranam Kolari, Akshay Java, Tim Finin, Tim Oates, and Anupam Joshi. Detecting Spam Blogs: A Machine Learning Approach. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI 2006)*, July 2006.

[113] Pranam Kolari, Akshay Java, Tim Finin, Tim Oates, and Anupam Joshi. Detecting Spam Blogs: A Machine Learning Approach. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI 2006)*. Computer Science and Electrical Engineering, University of Maryland, Baltimore County, July 2006.

[114] Risi Imre Kondor and John D. Lafferty. Diffusion kernels on graphs and other discrete input spaces. In *ICML '02: Proceedings of the Nineteenth International Conference on Machine Learning*, pages 315–322, San Francisco, CA, USA, 2002. Morgan Kaufmann Publishers Inc.

[115] William Krueger, Jonathan Nilsson, Tim Oates, and Tim Finin. *Automatically Generated DAML Markup for Semistructured Documents*. Lecture Notes in Artificial Intelligence. Springer, January 2004.

[116] Ravi Kumar, Jasmine Novak, Prabhakar Raghavan, and Andrew Tomkins. On the bursty evolution of blogspace. In *WWW*, pages 568–576, 2003.

[117] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew Tomkins. Trawling the Web for emerging cyber-communities. *Computer Networks (Amsterdam, Netherlands: 1999)*, 31(11–16):1481–1493, 1999.

[118] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew Tomkins. Trawling the web for emerging cyber-communities. *Computer Networks*, 31(11-16):1481–1493, 1999.

[119] Jim Lanzone. Which feeds matter? `http://blog.ask.com/2005/07/what_feeds_matt.html`, 2005.

[120] Andrew Lavallee. Friends swap twitters, and frustration - new real-time messaging services overwhelm some users with mundane updates from friends, March 16, 2007.

[121] Thomas Lento, Howard T. Welser, Lei Gu, and Marc Smith. The ties that blog: Examining the relationship between social ties and continued participation in the wallop weblogging system, 2006.

[122] Jure Leskovec and Christos Faloutsos. Sampling from large graphs. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 631–636, New York, NY, USA, 2006. ACM Press.

[123] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 177–187, New York, NY, USA, 2005. ACM.

[124] Jure Leskovec, Mary McGlohon, Christos Faloutsos, Natalie Glance, and Matthew Hurst. Cascading behavior in large blog graphs. In *SIAM International Conference on Data Mining (SDM 2007)*, 2007.

[125] David Liben-Nowell, Jasmine Novak, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. Geographic routing in social networks. *Proceedings of the National Academy of Sciences,*, 102(33):11623–1162, 2005.

[126] Yu-Ru Lin, Yun Chi, Shenghuo Zhu, Hari Sundaram, and Belle L. Tseng. Facetnet: a framework for analyzing communities and their evolutions in dynamic networks. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 685–694, New York, NY, USA, 2008. ACM.

[127] Yu-Ru Lin, Hari Sundaram, Yun Chi, Jun Tatemura, and Belle Tseng. Discovery of Blog Communities based on Mutual Awareness. In *Proceedings of the 3rd Annual Workshop on Weblogging Ecosystem: Aggregation, Analysis and Dynamics, 15th World Wid Web Conference*, May 2006.

[128] Yu-Ru Lin, Hari Sundaram, Yun Chi, Jun Tatemura, and Belle Tseng. Discovery of Blog Communities based on Mutual Awareness. In *Proceedings of the 3rd Annual Workshop on Weblogging Ecosystem: Aggregation, Analysis and Dynamics, 15th World Wid Web Conference*, May 2006.

[129] Bing Liu, Minqing Hu, and Junsheng Cheng. Opinion observer: analyzing and comparing opinions on the web. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 342–351, New York, NY, USA, 2005. ACM Press.

[130] Levon Lloyd, Prachi Kaulgud, and Steven Skiena. Newspapers vs blogs: Who gets the scoop? In *AAAI Spring Symposium on Computational Approaches to Analyzing Weblogs*, 2006. to appear.

[131] Craig Macdonald and Iadh Ounis. The trec blogs06 collection: Creating and analyzing a blog test collection. Technical report, 2006. Department of Computer Science, University of Glasgow Tech Report TR-2006-224.

[132] Craig Macdonald, Iadh Ounis, and Ian Soboroff. Overview of the trec-2007 blog track. In *The Sixteenth Text REtrieval Conference (TREC 2007) Proceedings*, 2007.

[133] Cameron Marlow. Audience, structure and authority in the weblog community. In *54th Annual Conference of the International Communication Association*, 2004.

[134] Adam Mathes. Folksonomies - cooperative classification and communication through shared metadata. `http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html`, 2004.

[135] S. Matsubara, S. Kimura, N. Kawaguchi, Y. Yamaguchi, and Y. Inagaki. Example-based Speech Intention Understanding and Its Application to In-Car Spoken Dialogue System. *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7, 2002.

[136] Chris McEvoy. Bloglines users are a load of knitters. `http://usability.typepad.com/confusability/2005/04/bloglines_user_.html`.

[137] Deborah L. McGuinness and Frank van Harmelen. Owl web ontology language overview. `http://www.w3.org/TR/2004/REC-owl-features-20040210/#s3.4`, 2004.

[138] Marjorie McShane, Sergei Nirenburg, Stephen Beale, and Thomas O'Hara. Semantically rich human-aided machine annotation. In *Proceedings of the Workshop on Frontiers in Corpus Annotations II: Pie in the Sky*, pages 68–75, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

[139] Marina Meila. Comparing clusterings–an information based distance. *Journal of Multivariate Analysis*, 98(5):873–895, May 2007.

[140] Marina Meila and William Pentney. Clustering by weighted cuts in directed graphs. In *SDM*, 2007.

[141] Stanley Milgram. The small-world problem, 1967.

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

[143] G. A. Mishne, K. Balog, M. de Rijke, and B. J. Ernsting. Moodviews: Tracking and searching mood-annotated blog posts. pages 323–324, 2007.

[144] G. A. Mishne, M. de Rijke, T. Salakoski, F. Ginter, S. Pyysalo, and T. Pahikkala. Language model mixtures for contextual ad placement in personal blogs. page 435446. Springer, Springer, 2006.

[145] G. A. Mishne and Natalie Glance. Predicting movie sales from blogger sentiment. 2006.

[146] Gilad Mishne. Autotag: a collaborative approach to automated tag assignment for weblog posts. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 953–954, New York, NY, USA, 2006. ACM Press.

[147] Gilad Mishne, David Carmel, and Ronny Lempel. Blocking blog spam with language model disagreement. In *AIRWeb '05 - 1st International Workshop on Adversarial Information Retrieval on the Web, at WWW 2005*, 2005.

[148] Gilad Mishne and Maarten de Rijke. Deriving wishlists from blogs show us your blog, and we'll tell you what books to buy. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 925–926, New York, NY, USA, 2006. ACM.

[149] B. Mohar. Some applications of laplace eigenvalues of graphs, 1997.

[150] I. A. Muslea, S. Minton, and C. Knoblock. Hierarchical wrapper induction for semistructured information services. *Journal of Autonomous Agents and Multi-Agent Systems*, 4(1/2):93–114, 2001.

[151] R. Nallapati and Cohen W. Link-PLSA-LDA: A new unsupervised model for topics and influence in blogs. In *Proceedings of the Second International Conference on Weblogs and Social Media*. AAAI Press, March 2008.

[152] Amit A. Nanavati, Siva Gurumurthy, Gautam Das, Dipanjan Chakraborty, Koustuv Dasgupta, Sougata Mukherjea, and Anupam Joshi. On the structural properties of massive telecom call graphs: findings and implications. In *CIKM '06: Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 435–444, New York, NY, USA, 2006. ACM Press.

[153] Bonnie A. Nardi, Diane J. Schiano, Michelle Gumbrecht, and Luke Swartz. Why we blog. *Commun. ACM*, 47(12):41–46, 2004.

[154] M. E. Newman. Scientific collaboration networks. i. network construction and fundamental results. *Phys Rev E Stat Nonlin Soft Matter Phys*, 64(1 Pt 2), July 2001.

[155] M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69:066133, 2004.

[156] M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69:066133, 2004.

[157] M. E. J. Newman. Power laws, pareto distributions and zipf's law. *Contemporary Physics*, 46:323, 2005.

[158] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74:036104, 2006.

[159] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks, August 2003.

[160] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, 2004.

[161] Kamal Nigam and Matthew Hurst. Towards a robust metric of opinion. In *Exploring Attitude and Affect in Text: Theories and Applications, AAAI-EAAT 2004*, 2004.

[162] Sergei Nirenburg and Victor Raskin. Ontological semantics, formal ontology, and ambiguity. In *FOIS '01: Proceedings of the international conference on Formal Ontology in Information Systems*, pages 151–161, New York, NY, USA, 2001. ACM Press.

[163] Sergei Nirenburg and Victor Raskin. Ontological semantics, 2005.

[164] Iadh Ounis, Maarten de Rijke, Craig Macdonald, Gilad Mishne, and Ian Soboroff. Overview of the TREC-2006 Blog Track. In *Proceedings of the 15th Text REtrieval Conference (TREC 2006)*, 2006.

[165] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[166] Gergely Palla, Imre Derenyi, Illes Farkas, and Tamas Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435:814, 2005.

[167] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of EMNLP 2002*, 2002.

[168] Chris Pirillo. Google: Kill blogspot already!!! `http://chris.pirillo.com/2005/10/16/`.

[169] Jason Pontin. From many tweets, one loud voice on the internet. *The New York Times*, April 22, 2007.

[170] Emanuele Quintarelli. Folksonomies: power to the people, 2005.

[171] Raghu Ramakrishnan and Andrew Tomkins. Toward a peopleweb. *Computer*, 40(8):63–72, 2007.

[172] E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, and A. L. Barabasi. Hierarchical organization of modularity in metabolic networks. *Science*, 297(5586):1551–1555, August 2002.

[173] P. Rayson and R. Garside. Comparing corpora using frequency profiling, 2000.

[174] Leisha Reichelt. http://www.disambiguity.com/ambient-intimacy/, March 2007.

[175] Matt Richardson and Pedro Domingos. Mining knowledge-sharing sites for viral marketing. In *KDD*, pages 61–70, 2002.

[176] R.V.Guha and Rob McCool. TAP: A semantic web toolkit. *Semantic Web Journal*, October 2003.

[177] Franco Salvetti and Nicolas Nicolov. Weblog classification for fast splog filtering: A url language model segmentation approach. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 137–140, New York City, USA, June 2006. Association for Computational Linguistics.

[178] Sam Gaddis Samuel D. Gosling and Simine Vazire. Personality Impressions Based on Facebook Profiles. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM 2007)*, 2007.

[179] Mark Sanderson and Bruce Croft. Deriving concept hierarchies from text. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 206–213, New York, NY, USA, 1999. ACM Press.

[180] David Schlangen, Manfred Stede, and Elena Paslaru Bontas. Feeding owl: Extracting and representing the content of pathology reports. In *RDF/RDFS and OWL in Language Technology: 4th ACL Workshop on NLP and XML*, July 2004.

[181] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA, 2004.

[182] Kaikai Shen and Lide Wu. Folksonomy as a complex network, Sep 2005.

[183] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[184] Xiaolin Shi, Belle Tseng, and Lada Adamic. Looking at the blogosphere topology through different lenses. In *ICWSM*, 2007. In submission.

[185] Xiaolin Shi, Belle Tseng, and Lada A. Adamic. Looking at the blogosphere topology through different lenses. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM 2007)*, March 2007.

[186] Masashi Shimbo, Takahiko Ito, and Yuji Matsumoto. Evaluation of kernel-based link analysis measures on research paper recommendation. In *JCDL '07: Proceedings of the 2007 conference on Digital libraries*, pages 354–355, New York, NY, USA, 2007. ACM.

[187] Clay Shirky. *Here Comes Everybody: The Power of Organizing Without Organizations*. The Penguin Press HC, 2008.

[188] David Sifry. State of the blogosphere, april 2006 part 1: On blogosphere growth. `http://www.sifry.com/alerts/archives/000432.html`.

[189] A. Smola and R. Kondor. Kernels and regularization on graphs, 2003.

[190] Xiaodan Song, Yun Chi, Koji Hino, and Belle Tseng. Identifying opinion leaders in the blogosphere. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 971–974, New York, NY, USA, 2007. ACM.

[191] Xiaodan Song, Yun Chi, Koji Hino, and Belle L. Tseng. Information flow modeling based on diffusion rate for prediction and ranking. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 191–200, New York, NY, USA, 2007. ACM.

[192] S.C. Sood, K.J. Hammond, S.H. Owsley, and L. Birnbaum. TagAssist: Automatic Tag Suggestion for Blog Posts. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM 2007)*, 2007.

[193] PF Strawson. Intention and Convention in Speech Acts. *The Philosophical Review*, 73(4):439–460, 1964.

[194] Zareen Syed, Tim Finin, and Anupam Joshi. Wikipedia as an Ontology for Describing Documents. In *Proceedings of the Second International Conference on Weblogs and Social Media*. AAAI Press, March 2008.

[195] Bill Tancer. *Click: What Millions of People Are Doing Online and Why it Matters*. Hyperion, 2008.

[196] Richard M. Tong. An operational system for detecting and tracking opinions in on-line discussion. In *SIGIR Workshop on Operational Text Classifiation*, 2001.

[197] Belle Tseng, Jun Tatemura, and Yi Wu. Tomographic Clustering To Visualize Blog Communities as Mountain Views. In *Proceedings of the 2rd Annual Workshop on Weblogging Ecosystem: Aggregation, Analysis and Dynamics, 15th World Wid Web Conference*, May 2005.

[198] Zeynep Tufekci. Can you see me now? audience and disclosure regulation in online social network sites, 2008.

[199] Zeynep Tufekci. Grooming, gossip, facebook and myspace: What can we learn about social networking sites from non-users. In *Information, Communication and Society*, volume 11, pages 544–564, 2008.

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

[201] Ulrike von Luxburg. A tutorial on spectral clustering. Technical report, Max Planck Institute for Biological Cybernetics, 2006. Technical Report No TR-149.

[202] Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schrödl. Constrained k-means clustering with background knowledge. In *ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning*, pages 577–584, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[203] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, June 1998.

[204] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, June 1998.

[205] Michael Witbrock, K. Panton, S.L. Reed, D. Schneider, B. Aldag, M. Reimers, and S. Bertolo. Automated OWL Annotation Assisted by a Large Knowledge Base. In *Workshop Notes of the 2004 Workshop on Knowledge Markup and Semantic Annotation at the 3rd International Semantic Web Conference ISWC2004*, November 2004.

[206] Qianjun Xu, Marie desJardins, and Kiri Wagstaff. Active constrained clustering by examining spectral eigenvectors. In *Discovery Science*, pages 294–307, 2005.

[207] Lan Yi and Bing Liu. Web page cleaning for web mining through feature weighting. In *Proceedings of Eighteenth International Joint Conference on Artificial Intelligence, IJCAI-03*, 2003.

[208] Lan Yi, Bing Liu, and Xiaoli Li. Eliminating noisy information in web pages for data mining. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD-2003*, 2003.

[209] Dengyong Zhou, Christopher J. C. Burges, and Tao Tao. Transductive link spam detection. In *AIRWeb '07: Proceedings of the 3rd international workshop on Adversarial information retrieval on the web*, pages 21–28, New York, NY, USA, 2007. ACM Press.

[210] Ding Zhou, Jiang Bian, Shuyi Zheng, Hongyuan Zha, and C. Lee Giles. Exploring social annotations for information retrieval. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 715–724, New York, NY, USA, 2008. ACM.