2007

Profiling topics on the Web for knowledge discovery

Aditya Kumar Sehgal
University of Iowa

PROFILING TOPICS ON THE WEB FOR KNOWLEDGE DISCOVERY

by

Aditya Kumar Sehgal

An Abstract

Of a thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy degree
in Computer Science in the Graduate College of
The University of Iowa

December 2007

Thesis Supervisor: Professor Padmini Srinivasan

ABSTRACT

The availability of large-scale data on the Web motivates the development of automatic algorithms to analyze topics and to identify relationships between topics. Various approaches have been proposed in the literature. Most focus on specific topics, mainly those representing people, with little attention to topics of other kinds. They are also less flexible in how they represent topics.

In this thesis we study existing methods as well as describe a different approach, based on profiles, for representing topics. A Topic Profile is analogous to a synopsis of a topic and consists of different types of features. Profiles are flexible to allow different combinations of features to be emphasized and are extensible to support new features to be incorporated without having to change the underlying logic.

More generally, topic profiles provide an abstract framework that can be used to create different types of concrete representations for topics. Different options regarding the number of documents considered for a topic or types of features extracted can be decided based on requirements of the problem as well as the characteristics of the data. Topic profiles also provide a framework to explore relationships between topics.

We compare different methods for building profiles and evaluate them in terms of their information content and their ability to predict relationships between topics. We contribute new methods in term weighting and for identifying relevant text segments in web documents.

In this thesis, we present an application of our profile-based approach to explore social networks of US senators generated from web data and compare with networks generated from voting data. We consider both general networks as well as issue-specific networks. We also apply topic profiles for identifying and ranking experts

given topics of interest, as part of the 2007 TREC Expert Search task.

Overall, our results show that topic profiles provide a strong foundation for exploring different topics and for mining relationships between topics using web data. Our approach can be applied to a wide range of web knowledge discovery problems, in contrast to existing approaches that are mostly designed for specific problems.

Abstract Approved: _____
Thesis Supervisor

_____
Title and Department

_____
Date

PROFILING TOPICS ON THE WEB FOR KNOWLEDGE DISCOVERY

by

Aditya Kumar Sehgal

A thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy degree
in Computer Science in the Graduate College of
The University of Iowa

December 2007

Thesis Supervisor: Professor Padmini Srinivasan

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

PH.D. THESIS

This is to certify that the Ph.D. thesis of

Aditya Kumar Sehgal

has been approved by the Examining Committee
for the thesis requirement for the Doctor of
Philosophy degree in Computer Science at the
December 2007 graduation.

Thesis Committee: _____
Padmini Srinivasan, Thesis Supervisor

_____
Alberto Segre

_____
David Eichmann

_____
James Cremer

_____
Lisa Troyer

To my family

ACKNOWLEDGMENTS

I came to this wonderful country seven and a half years ago to pursue my graduate studies. During this time I have come in contact with a number of people, all of whom have touched me in one way or another. Some of these people I thank below while others, who are not named, I also am grateful to.

This dissertation would not have been possible without the support and guidance of my advisor, Prof. Padmini Srinivasan. She has been my teacher, my mentor and my friend. She introduced me to the field of text mining and was always there to answer my questions and encourage me. Her guidance has helped me become the professional I am today. I have learnt a great deal from her both on the professional front and on the personal front. Being her Research Assistant for the better part of 3 years also helped me develop my research aptitude and provided me with the opportunity to publish papers.

I am grateful to members of my dissertation committee, Profs. Jim Cremer, Alberto Segre, Dave Eichmann and Lisa Troyer, for their questions, comments, ideas and suggestions. Discussions with Prof. Segre, Lisa and Dave at different points during the research phase of my dissertation proved very useful. I am additionally grateful to Prof. Cremer for backing me as Chair of the department ever since I arrived at the University of Iowa in 2001. I am also additionally grateful to Prof. Segre for his support as Associate Chair of the department and for always showing an interest in how things are going for me. I would also like to thank my teachers and my professors throughout my academic years.

Some parts of this thesis were presented at the I3 workshop at the 2007 WWW Conference in Canada. I would like to thank the reviewers for their feedback.

I am grateful to my colleague Xin Ying Qiu for her interest and encouragement.

She was also a collaborator in some of the research projects I worked on. I would also like to thank my supervisor and my colleagues at Parity Computing for their understanding and good wishes.

During my stay in Iowa City I have been fortunate to have become friends with many wonderful people. I am especially grateful to Chetan, Dada and Rajeev for being willing to listen to my ramblings about my research and help me gain clarity on certain issues during times of confusion.

I would like to thank my extended family, my father-in-law, mother-in-law, Mudit Bhaiya, Sarika, Ashish Bhaiya, and Somna Bhabhi for their love and encouragement.

My brother Anu has always been a source of love and support. His intelligence and smartness are a constant source of motivation for me to do better things. I am also grateful to my sister-in-law Naina for her caring and encouragement.

I owe an unending debt of gratitude to my wife Varsha. During the last five years, she has been my companion through thick and thin. I am thankful for her love, encouragement and patience. She has had to make so many sacrifices to ensure that I could focus on my work and succeed. Through many a difficult time her support was what carried me through.

Finally, I would like to thank my parents. They instilled in me a sense of independence and self reliance at an early age and taught me the value of sincerity and hard work. These qualities have been of tremendous importance to me during my life and especially during my Ph.D. Without their love and guidance, I would not be where I am today. I know I can always count on their support in whatever I do.

Mom, Dad, Varsha, Anu and Naina – this is for you!

ABSTRACT

The availability of large-scale data on the Web motivates the development of automatic algorithms to analyze topics and to identify relationships between topics. Various approaches have been proposed in the literature. Most focus on specific topics, mainly those representing people, with little attention to topics of other kinds. They are also less flexible in how they represent topics.

In this thesis we study existing methods as well as describe a different approach, based on profiles, for representing topics. A Topic Profile is analogous to a synopsis of a topic and consists of different types of features. Profiles are flexible to allow different combinations of features to be emphasized and are extensible to support new features to be incorporated without having to change the underlying logic.

More generally, topic profiles provide an abstract framework that can be used to create different types of concrete representations for topics. Different options regarding the number of documents considered for a topic or types of features extracted can be decided based on requirements of the problem as well as the characteristics of the data. Topic profiles also provide a framework to explore relationships between topics.

We compare different methods for building profiles and evaluate them in terms of their information content and their ability to predict relationships between topics. We contribute new methods in term weighting and for identifying relevant text segments in web documents.

In this thesis, we present an application of our profile-based approach to explore social networks of US senators generated from web data and compare with networks generated from voting data. We consider both general networks as well as issue-specific networks. We also apply topic profiles for identifying and ranking experts

given topics of interest, as part of the 2007 TREC Expert Search task.

Overall, our results show that topic profiles provide a strong foundation for exploring different topics and for mining relationships between topics using web data. Our approach can be applied to a wide range of web knowledge discovery problems, in contrast to existing approaches that are mostly designed for specific problems.

TABLE OF CONTENTS

LIST OF TABLES

Table

LIST OF FIGURES

Figure

CHAPTER 1
INTRODUCTION

It is generally agreed that no source of information today compares to the World Wide Web in terms of sheer size and diversity. There is also broad recognition of the tremendous opportunity for mining and knowledge discovery from the Web. However, web mining research is at an early stage, especially with regards to knowledge discovery. Far less has been achieved compared to text mining in specialized domains, thus offering a tremendous opportunity for further research in Web mining.

In this thesis we focus on the goal of using data from the Web for knowledge discovery. Various approaches have been proposed in the literature. However, most focus on specific topics, mainly people entities. There is little research exploring approaches capable of handling different kinds of 'topics' beyond entities. Existing approaches also make somewhat arbitrary choices regarding some key factors such as the number of web pages to use for mining and knowledge discovery, how much data in each page to use, methods for extracting features from web pages, and methods for weighting the extracted features.

We study existing methods as well as describe a different approach, based on profiles, for representing general topics. A Topic Profile is analogous to a synopsis of a topic and consists of different types of features extracted from a set of relevant pages. Topic profiles are flexible in that they allow different combinations of features to be emphasized depending upon the knowledge discovery goals. They are extensible in that new types of features can be easily incorporated into a profile without having the make any modifications to the underlying logic. Different options such as the number of relevant web pages or the number and types of features extracted can also be decided based on the requirements of the problem as well as the characteristics of the data. Most significantly, topic profiles provide a framework to explore relationships

between topics, an aspect explored in this thesis.

In sum, topic profiles provide an abstract framework that can be used to create different types of concrete representations for all kinds of topics as well as for exploring different types of relationships between topics.

In this thesis we compare different approaches for building profiles and evaluate them in terms of their information content and ability to predict relationships between topics. We contribute new methods in term weighting and for identifying relevant text segments in web documents. We also apply our profile-based approach to explore social networks of senators and to the task of identifying experts given topics of interest. A by-product of this thesis is that we provide an evaluation of some standard foundational tools used in mining general text documents. Specifically we evaluate their effectiveness when applied to web documents.

1.1 Thesis outline

In chapter 2 we provide the necessary background in text and web mining. In chapter 3 we make several observations regarding the field in general as well as specific existing approaches. These observations provide the motivations behind the work in this thesis. In chapters 4 and 5 we first introduce our approach of topic profiles and then describe in detail the methodology we follow to build profiles from text. We also describe a web-based implementation for topic profile building called WebKD. In chapters 6 and 7 we describe a two-pronged evaluation process. First, we evaluate different methods for building profiles on the basis of their information content. Specifically, we compare profiles generated from web data with profiles generated from Wikipedia. Second, we evaluate different types of profiles on the basis of their ability to predict relationships, specifically protein interactions, using web data. This addresses a secondary goal of this thesis, which is to explore the extent to which the heterogeneous Web may be used to support knowledge discovery

in a specialized domain such as biomedicine. In chapter 8 we present an exploratory research with US senator networks generated from web data, comparing these with networks generated from vote data. In chapter 9 we describe our participation in the 2007 TREC Expert Search task using our profile-based approach. Finally, in chapter 10 we provide a summary and general discussion and also outline avenues for future research.

CHAPTER 2
BACKGROUND

2.1 Text Mining

Text mining also known as Text Data Mining (TDM) [72], Knowledge Discovery in Textual Databases (KDT) [56] and Literature-based Discovery (LBD) [126] can be described as the process of identifying *novel* ideas from a collection of texts (also known as a corpus). By novel we mean information that is not explicitly present in the text source being analyzed. The kinds of ideas of interest are those indicating associations, hypotheses, trends, etc. This view of text mining is consistent with the definition proposed by Hearst in her highly cited paper [72]. To illustrate, consider the research of Swanson [125] with *Raynauds Disease* and *Fish Oils*. Swanson was interested in Raynauds Disease and read a number of research papers on the subject. He observed that Raynauds was exacerbated by certain factors such as *platelet aggregability*, *vasoconstriction*, and *blood viscosity*. From independent literature he also observed that these factors were mitigated by *fish oils*. Putting the two together he postulated that fish oils may be beneficial for Raynauds. This association was unknown at the time and was later confirmed by bioscientists.

In our research we agree with Hearst's view that novelty with respect to the text collection is a requirement in text mining. However, like many others [133, 83] we adopt a more flexible definition of what constitutes novelty. Specifically, we see a subjective dimension in what is or is not perceived to be novel. Although not necessary, text mining efforts tend to adopt a multi-document perspective, with novel associations inferred by combining evidence from more than one document. Given the large amount of information available in text form today, we believe that tools that automatically find interesting relationships, hypotheses or ideas, or assist the user

in finding these will be extremely useful. Interestingly, most of the existing research in text mining has been limited to the context of biomedicine, part of which can be attributed to the early efforts of Swanson and Smalheiser [125, 128, 129, 130, 131].

Text mining is an inter-disciplinary field using techniques from the fields of information retrieval [119], natural language processing [115], machine learning [118], visualization [75], clustering [143], and summarization [99], among others. Text mining represents a significant step forward from text retrieval. It is a relatively new and vibrant research area that is changing the emphasis in text-based information technologies from low level 'retrieval' & 'extraction' to higher level 'analysis' & 'exploration' capabilities. In recent years there has been increased interest in text mining. One can see more papers in the area being published in top conferences such as SIGIR [1] and the WWW conference [12]. Also, of late, there has been a proliferation of text mining workshops [13, 5, 20].

2.1.1 Application Domains

There are many domain specific text collections available electronically. We have for example, MEDLINE [9], Reuters newswire data [19], SEC filings of companies [11], archives of mailing lists dealing with specific subject areas and collections of customer emails, product reviews etc. These domain specific corpi motivate the design of customized text mining algorithms that can exploit domain knowledge to provide better performance than generic text mining algorithms. Three of the most popular domains, where text mining techniques are being actively developed and used, are Biomedicine (including Bioinformatics), Business Intelligence and Counter-Terrorism.

2.1.1.1 Biomedicine

The biomedical research literature is a very promising target for text mining. Given the extensive presence of biomedical papers in digital form, as well as their formal and technical vocabulary, they offer a profitable area for automatic text mining. Moreover the high level of interest in biotechnology has made it one of the most active application domains for text mining. In fact a recent paper [33] in *Nature* coins the term 'conceptual biology' for the science of text mining in biology while describing its value in fueling progress in bioinformatics.

A significant portion of text mining research in this domain has been done in the context of MEDLINE, an online database of over 15 million records representing the published literature in biomedicine from the 1960s onwards. Each MEDLINE record consists of a title, an abstract, a set of manually assigned metadata terms (known as MeSH terms), and several other fields. The huge and growing size of biomedical research makes it almost impossible for someone to keep abreast of all the literature in their domain. Also, given the inter-disciplinary nature of research, one needs to keep track of related fields apart from one's own field. This further underlines the challenge in biomedical research. Therefore, tools that filter through the literature and retrieve relevant papers are highly valued. But looking beyond retrieval, tools that help in discovering new relationships and suggesting hypotheses from the literature have enormous potential.

A particular sub-problem in bioinformatics that has received a fair amount of attention from text mining researchers is gene/protein analysis. This is partly due to the large amount of literature on genes and proteins, which consequently has led to a high level of interest in genomic research. Automatic extraction of gene and protein names in text [140] is an important part of this research. A key motivation is that once these entities are identified, it will become easier for scientists to connect the

information available in MEDLINE with that in allied databases such as LocusLink [15], OMIM [17] and SwissProt [10]. What makes this task challenging is the inherent ambiguity associated with gene/protein nomenclature. Dealing with synonymy and homonymy with respect to gene and protein names is part of this challenge. In other research (independent of this thesis and published in *BMC Bioinformatics*) we have proposed and tested methods to tackle this problem [113]. Our methods are designed from a retrieval perspective. Other solutions proposed for identification of gene and proteins in text include, for example, machine learning -based approaches [145, 70] and hidden markov model -based approaches [90, 96]. In addition to genes/proteins, there are numerous efforts on identification of other entities such as organs, cells, biological pathways, etc., from text.

Strictly speaking the entity identification problem, described above, is an example of information extraction and not text mining. However, we intentionally refer to this body of research as it is a fundamental problem that seriously impacts higher level text mining capabilities. Operating on top of such extraction efforts, we observe the mining of 'higher level' information. In this thesis on web mining through profiles, we face similar challenges in extracting names of different entities from web documents.

Continuing with the theme of mining with genes and proteins, we have for example, Jenssen et al., who created PubGene [75], a network of genes, which can be used for mining functional relationships and for gene expression analysis. In their network, two genes are connected if they co-occur in the title or abstract of a MEDLINE document. They build their network over 13,000 human genes mentioned in public databases such as HUGO [23], LocusLink [15], the Genome Database [22], and GENATLAS [25]. Other mining efforts have identified functional relationships between genes [116, 122], protein-protein interactions [92, 34], and interactions between

genes and gene products [115].

A more general approach to discover novel biomedical pathways was developed by Swanson [125]. The general idea is that two concepts A and C are potentially connected if A co-occurs in some document with some concept B, and B co-occurs in some document with C. This implication-based or transitive discovery process was successfully used by the authors to discover several novel relationships such as between *Raynauds disease* and *fish oils* [125], and *migraine* and *magnesium* [128] among others [129, 130, 131]. Swanson along with his colleague Smalheiser essentially designed two kinds of discovery processes that were later named 'Open' and 'Closed' discovery [139]. Ultimately we seek to explore these types of discovery strategies on web data.

2.1.1.2 Business Intelligence

A major concern of any business is to minimize the amount of guesswork involved in decision making and thereby reduce risk. Most data mining techniques, such as association rule mining and data warehousing, were originally created to help remove the uncertainty or alleviate it, so that decision making could be more sound. However, data mining can help only upto a certain point, since the majority of data available with a company (reports, memos, emails, planning documents, etc.) is in the form of text. Since text is not structured enough for data mining techniques to apply, text mining holds promise. For example, text mining techniques, built by combining methods for feature selection, clustering and summarization, allow business professionals to extract important words/patterns from documents, group related documents together, read only summaries and drill down to the full documents as necessary, thereby saving precious time and effort. Data mining and text mining techniques can also complement each other. For example, data mining techniques may

be used to reveal the occurrence of a particular event while text mining techniques may be used to look for an explanation of the event. Text mining can also be used to identify implicit connections, wherein lies its, for the most part *untapped*, potential value for businesses.

Research in the application of text mining techniques to the business area is encouraging. In [32] Bernstein et al. analyze co-occurrence based association rules that relate different companies. Their analysis is done on over 22,000 business news stories. They begin with an information extraction software, *ClearForest* [3], to extract the set of company names from the text. As shown later we use this software as well. Bernstein et al. then use disambiguation techniques on this set to identify all the unique company names. For example, H.P. and Hewlett Packard are merged. A graph structure is used to visualize the processed data. Each node in the graph represents a company and an edge represents a co-occurrence based association between two companies. To eliminate random associations they link two companies only if the strength of their association is above a minimum support threshold. From this graph they identify *hubs*, which represent dominant companies in different industries. They also use the vector space model (from IR) to represent companies as weighted link vectors. They consider the cosine similarity score between a company vector and the average industry vector as an estimate of the relatedness of a company to its industry. Additionally, the similarity between different average industry vectors gives a measure of how closely related the industries are to each other. For example, not surprisingly, they found that the computer software industry and the computer hardware industry vectors were fairly similar. Although this research did not reveal any new knowledge, as acknowledged by the authors, such methods may be useful for knowledge discovery in other areas. In a later chapter (chapter 8) we also generate and analyze graphs depicting relationships, specifically between US senators.

In [77] Gerdes describes *EDGAR-Analyzer*, a text mining tool that analyzes the free-text portion of records in the *EDGAR* database[1], maintained by the Securities and Exchange Commission (SEC). *EDGAR* consists of financial and operational disclosures of public companies. This tool allows a user to specify subject areas of interest, which it then uses to extract relevant concepts from the text, backed up by the actual text passages that contain those concepts. This kind of analysis can help in monitoring key company characteristics, which may then be used by investors for making investment decisions. Of particular interest is a case study in the paper wherein Gerdes uses his methods to explore, by mining company filings, the different extents to which companies were prepared for the Y2K problem at the end of the last century.

2.1.1.3 Counter-Terrorism

The use of text mining techniques in helping counter-terrorism efforts is a relatively recent effort. Government agencies are investing considerable resources in the surveillance of all kinds of communication, including email. Since time is critical and given the scale of the problem, it is infeasible to monitor email manually. It is imperative for security agencies to be able to analyze large amounts of text quickly and accurately, and also understand the implicit connections between various sources of information. Thus automatic text mining offers considerable promise. An example of an operational text mining system is COPLINK [46]. Developed at the University of Arizona and currently being used by local police there and in several states, this system identifies novel connections between criminals from information across multiple text databases that are maintained by different agencies.

The Echelon network [8], run by a conglomerate of English speaking countries,

[1] http://www.sec.gov/edgar.shtml

is one of the largest surveillance systems in the world. The system can capture various communication signals such as radio, telephone, faxes and emails from nearly anywhere in the world. An estimated 3 billion messages are intercepted daily. The massive amount of data are analyzed manually and by computer programs to detect interesting patterns. However, the size of the data makes it impossible to do a complete sweep and thus analysts must know beforehand what they are looking for to extract any intelligence.

In addition to biomedicine, Swanson et al. [132] have also applied their discovery methods to this domain. They mine the literature for viruses with as yet unrecognized potential for use as biological weapons. They essentially partition the literature on viruses, in MEDLINE, into two parts. The first part consists of documents that talk about the genetic aspects of virulence, and the second part consists of documents that talk about the transmission of viral diseases. They assume that a virus that can be used as a biological weapon would have both these properties. They then create a list of virus terms extracted from both of these sets. Most of the viruses already recognized as potential biological weapons are present in this list. They hypothesize that since the other viruses in the list share important properties with the known biological agents, they are also likely to be potential biological weapons.

2.2 Web Mining

Our focus in this thesis is on Web Mining, which can be thought of as an extension of text mining to the Web. As with text mining, there exists an entire spectrum of opinions in the literature as to what constitutes web mining. At one extreme, some authors term standard data mining research such as classification, clustering and information extraction, applied to the Web, as web mining. For example, Sun

et al. [124] and Yao & Choi [142] include web page classification and web page clustering tasks under web mining, respectively. At the other extreme, some authors only include knowledge discovery, i.e., identification of novel information through inference mechanisms, as web mining. For example, Gordon et al. [67] predict novel relationships from web data such as between *genetic algorithms* and *cancer detection*, which are not explicitly mentioned in any web document. Similarly Ben-Dov et al. [31] apply a knowledge discovery approach to hypothesize a relationship between two people, which is again not explicitly mentioned in any document. We represent the broad spectrum of viewpoints in our review of web mining.

Although the Web is a mix of various types of documents such as audio, video, images, etc., web mining research typically focuses on manipulating text. It is generally thought that most of the Web is composed of text data. This data is available in the form of structured documents (automatically generated pages), semi-structured documents (html pages, etc.), and free-text documents (text files).

The nature of the Web brings different kinds of challenges to the forefront, such as the verification of extracted facts, as well as the reliability of any discovered novel information. Some of these challenges appear greater than for domain specific corpi. For instance, the likelihood of false positive relations being identified on the Web is possibly greater than when mining a specialized corpus like MEDLINE. Resolving ambiguity is also a greater challenge as a term can have far more meanings, under different domains. E.g., *Matrix* is both a movie and a mathematical object, *Cricket* is both a sport and an insect. Ambiguity in people names is also quite common on the Web. Also, given the loose sense of control on the Web, web mining, to a great degree, depends on filtering to eliminate low quality information (e.g., [144]).

Metadata elements, even those that are actively promoted, such as, the Dublin Core Metadata Element Set (DCMES) [7] (E.g., *Type*, *Creator*), are seldom seen, or

used inconsistently. In this context, Natural Language Processing (NLP) techniques that allow parsing of text data and that attach semantics to words, by named entity tagging, assume significance. There is extensive NLP research on key problems such as word sense disambiguation, part-of-speech tagging, phrase identification, extraction of relations, etc., especially in specialized domains. For example, in [115] Sekimizu et al. extract relationships between gene products and proteins from MEDLINE by identifying subject and object terms for frequently seen verbs such as *activate* and *interact*. The application of NLP methods to web data presents additional challenges. Semi-structured web documents differ from raw text documents typically used in Information Retrieval (IR)research, in terms of content and presentation style and contain additional structures such as tables and images. It has also been reported that in general web documents tend to follow a different set of linguistic rules [103]. These factors can affect the performance of standard NLP methods on web data [28]. Often, web NLP methods use additional features, such as the tag structure, in semi-structured documents [42, 103]. Machine learning techniques have also been applied, for example, to learn information extraction rules for semi-structured and unstructured text [118] and learn hidden markov models that assign semantic tags to tokens in web documents [110].

One approach to organize the research literature on web mining is on the basis of the type of web data being considered [89, 82]. This approach yields three categories, Web Content Mining, Web Structure Mining, and Web Usage Mining. However, these categories are not rigid and the literature often reveals hybrid approaches. This is especially true for content mining and structure mining. We describe each of these categories in more detail below.

2.2.1 Web Content Mining

As the name suggests, these techniques utilize the content within web pages. Content includes textual information, tags, figures, tables, etc. Web content mining relies on strategies to represent the content (and sometimes the structures) in web documents. Typically this is done from either a database perspective (DB) or an information retrieval (IR) perspective [82].

From a DB perspective, relational structures are used to model and manage information in structured and semi-structured web documents. This allows retrieval of information using more sophisticated queries than plain keyword queries. For example, WebSQL [94] uses a virtual schema for HTML documents based on tag attributes and provides support for queries such as *SELECT d.url, d.title FROM Document d SUCH THAT d MENTIONS "aluminum"*. Another example is the W3QS system developed by Konopnicki and Shmueli [81], which views the Web as a large database and provides an SQL-like interface for querying both content and structural information.

From an IR perspective, the free-text in web documents, are modeled using the unordered *bag-of-words* approach [98, 76]. This emphasizes the importance of individual words and/or phrases using statistically derived weights. Additionally, the tags in semi-structured documents (e.g., HTML and DHTML) may convey certain kinds of semantic information and can influence the bag of words representation. For example, the <h1> tag in an HTML document may be interpreted as highlighting 'important' phrases in the document. There are efforts [136, 124] that use the tag structure, in addition to words and phrases, to model web documents. In many instances, as described below, hyperlinks are also considered as part of the content. This cross content and structure mining allows for jointly using text and link features in web documents.

Not surprisingly, many popular data mining techniques, such as classification,

clustering, and pattern mining, have been applied to the web context. For example, Sun et al. [124] use a support vector machine (SVM)-based algorithm to categorize web pages into different classes. They use both text features and contextual features derived from the HTML tags of the pages. Joachims [76] also considers the problem of classifying web pages. He too uses an using a SVM-based classifier but with a composite kernel. This is a combination of two kernels, one for the text features in the documents and the other for link features. Yao and Choi [142] describe a bidirectional hierarchical clustering approach to cluster web documents. Their approach works in two phases. In the bottom-up phase they aim to maximize intra-cluster similarity and in the top-down phase they aim to minimize the inter-cluster similarity. In [107], Ravichandran and Hovy describe an approach for automatically answering certain types of questions from patterns in web documents. They automatically generate a set of patterns for each question type via a bootstrapping process. They test their approach using questions from the TREC[2]-10 QA track [21] and extract answers from both the TREC-10 corpus and the Web. They observe that in general, the answers extracted from the Web are more accurate. They attribute this to the abundance of data on the Web, which makes it more likely for their patterns to match with phrases/sentences containing the correct answer.

Looking beyond the many applications of standard data mining techniques to solve web-based problems, we see 'knowledge discovery' efforts, though in comparison these are rather few. There are efforts similar to Swanson and Smalheiser's strategies (described in the previous chapter). Gordon et al. [67] apply open and closed discovery to identify novel connections between entities in web documents. They perform two types of experiments. Firstly they use open discovery to find new applications for existing techniques. E.g., they hypothesize the potential application of *genetic algorithms* in *cancer detection* as well as *financial modeling*. Secondly, they use closed

[2]Text REtrieval Conference

discovery to find entities related to an entity of interest, via manually specified intermediate entities. E.g., they hypothesize connections between *genetic algorithms* and *adaptive mesh* as well as *filter algorithm*.

Ben-Dov et al. [31] use a similar approach to extract implicit connections in the Web. Firstly, they use information extraction methods to identify interesting concepts in the text. Next, they identify concept pairs that are explicitly related via co-occurrence in the same sentence. Finally, they apply a transitive process (akin to open discovery) to connect two unrelated concepts via intermediate concepts. E.g., they hypothesize a 'connection' between *Pope John Paul II* and *Osama Bin Laden* via *Ramzi Yousef* (who attempted to assassinate the Pope) as he is explicitly connected to both in the literature. This relationship is novel as no sentence in the text mentions both the Pope and Bin Laden. In [26], Adamic predicts relationships between people on the Web using features present on their home pages. An individual is characterized by the text on his/her page, the links to and from the page, and the mailing lists he/she subscribes, which are mentioned on the page. People with similar characteristics and no direct link between their home pages are hypothesized to be related. Our work on senator relationships (chapter 8) is related to this background literature.

2.2.2 Web Structure Mining

Web structure mining involves the use of hyperlinks between web pages. Since the Web can be thought of as a graph, graph theoretic methods have become quite popular. Typically in the literature, the nodes of the graph represent individual web pages and edges represent hyperlinks between the web pages. Although not referred to as knowledge discovery in the classical sense, i.e., as per the Swanson and Smalheiser approach, this can also be used for hypothesis discovery.

It is well known that the link structure of the Web has been used to improve web searching [39, 44]. However, many other link-based applications are oriented more towards knowledge discovery. The whole area of web community discovery falls under this category. For example, Flake et al. [60] describe an approach to identify web communities. Their start with a few seed pages and then expand the community by crawling, using a focused crawler, upto a fixed depth. They then use the min/max flow cut framework to produce a cut between the source, which consists of pages in the community, and the sink, which consists of pages outside the community. Kumar et al. [84] concentrate on identifying communities on the Web that are not yet mainstream or popular. They define a community as a bipartite graph, termed a *core*, which consists of *fans* or pages that link to other pages to in the community and *centers* or pages that are linked to by other pages in the community. They analyze a dataset of over 200 million web pages and use various pruning techniques to reduce the size of the data. They extract those communities that have fans with a minimum number of outlinks and centers with a minimum number of inlinks. These communities are evaluated manually. They also perform temporal analysis and evaluate the survival rate of the communities they extracted.

Golbeck and Hendler [66] focus on trust in web-based social networks. Specifically, they analyze how trust ratings can be propagated along the edges in a person network. This allows for trust to be inferred (mined) between people who are not necessarily directly connected. They use this concept to design an email client that filters email on the basis of direct and inferred trust ratings.

2.2.3 Web Usage Mining

In contrast to web content mining and web structure mining, research in web usage mining focuses on the secondary data generated from user interactions with

the Web. Secondary web data sources include server access logs, proxy server logs, browser logs, user profiles, registration data, cookies, user queries, etc. These data can be used to model user behavior or learn user profiles [141], as well as learn user navigational patterns on the Web, which can then be used improve web site design and personalize web sites [95], among others.

Tan and Kumar [134] describe a hybrid approach that focuses on usage data and links to identify indirect associations between pages on a web site, via intermediate pages. Their goal is to analyze user navigational behavior and facilitate improvements in the organization of pages on a web site. Pages are indirectly associated via an intermediate ordered sequence of pages, called a *mediating sequence*. They describe a two-phased approach. First, they use a standard frequent itemset generation technique, such as the Apriori algorithm [27], to identify frequently accessed sequences of pages in the usage data. Second, sequences are iteratively joined to produce candidate indirect associations between pages. Two sequences are joined only if they have a certain number of pages in common. An indirect association between two pages a and b is considered a potential candidate only if they are not frequently associated, both a and b are frequently associated with pages in the mediating sequence, and there is a dependence relationship between both a and b and the pages in the mediating sequence. They consider three types of indirect associations, viz. Convergence, Divergence and Transitivity. a and b are related via convergence if they both link to the first page of the mediating sequence. They are related via divergence if they are both linked to by the last page of the mediating sequence. Transitivity implies that a points to the first page in the mediating sequence and b is linked to by the last page in the mediating sequence. Candidate indirect associations between pages are evaluated manually.

In conclusion, the drive to solve problems in specialized areas including biomedicine,

business intelligence, and counter-terrorism provides a key motivation for the development of future text mining techniques. Most of the existing research has been done in the context of biomedicine, part of which can be attributed to the early efforts of Swanson and Smalheiser. While the discovery of novel ideas or associations is the main focus of text mining, it relies significantly on methods from core areas such as information extraction, text retrieval, and inferencing methods. In the next chapter we present some key observations derived from our review of the literature and motivate our own research.

CHAPTER 3
OBSERVATIONS AND MOTIVATIONS

We now make several observations that derive from our review of the text mining and web mining research. These observations provide the key motivations for our research. As with text mining, a wide spectrum of goals and methods are observed under the umbrella of web mining. Many researchers consider standard data mining methods such as classification, clustering, pattern matching, etc., applied to the Web, as web mining. However, our interest and focus is on knowledge discovery. Thus, we differentiate between efforts that seek to retrieve, classify, and cluster web pages from efforts that seek to infer new ideas in the form of associations. Our long-term goal is to discover novel hypothesis, typically in the form of associations between topics/entities (such as people, organizations, etc.) by exploiting relationships expressed in web data. In addition we are interested in the discovery of novel relationships such as those reflected by online communities and social networks.

We find extensive research on knowledge discovery in specialized domains, especially in biomedicine. However, we observe very few comparable efforts that operate off the Web. Thus our first goal is to contribute a systematic exploration of knowledge discovery methods designed for web data.

Of the few instances of web mining research that exist, most [67, 31] apply standard methods from other domains, such as Swanson and Smalheiser's Open and Closed discovery approaches. Others [26, 106] define relationships directly based on similarity between ad-hoc representations of entities. These methods do not take advantage of additional information found in web documents such as tags and hyperlinks. Thus, we are interested in contributing to research in knowledge discovery that extends methods developed in specialized domains and utilizes the special features of the Web.

There exists a substantial body of research devoted to the identification of online communities (e.g., [60, 84]) and social networks (e.g., [78, 93]) from the Web. However, we observe that most methods are constrained to particular explicit indicators, such as shared hyperlinks or co-occurrence, to infer relationships between entities. Far less research exists on identifying and analyzing networks where relationships are more indirect.

Our analysis of the literature reveals that most web mining efforts (e.g., [26, 106, 31]) focus on specific topics, mostly people entities. The focus is seldom on all kinds of topics. A principal goal in our web mining research is to facilitate analysis of web data relevant to any topic of interest as well as to identify (or predict) relationships between any kind of topics.

We define a topic as any subject of interest to a user. Examples include *Bill Clinton*, *A1BG Gene*, *Rainfall in the United States*, and *Cancer in Children*. Observe that while the first two are also entities, the latter two are not and are more general. Topics may also be identified by other types of text units such as by one or more sentences. One may also regard any web search query as implicitly representing an underlying topic. Our interest is in web mining methods that are not constrained to particular varieties of topics. As an example, given an arbitrary group of topics, we would like to identify links between them and explore relationships.

Another observation that motivates our research is regarding the differences between various web mining approaches along two major dimensions as depicted in figure 3.1. The first dimension represents the number of web pages used to represent a topic. This can range from a single page, such as a home page, to any number of pages retrieved from a search engine. The second dimension refers to the portions of a web page (we refer to this as level of data) on a given page from which features descriptive of the topic are extracted. Two options are repeatedly used in the literature. First

is to use information available at the instance level. This is the text including and surrounding an individual mention of a topic (more specifically an entity) in a page (e.g., [67, 31]). Thus, *Bill Clinton* would be represented by the appearance of the phrase 'Bill Clinton' or 'President Clinton' and a window of words surrounding these phrases. The second approach is to use the information in a single web page, usually the home page (e.g., [26]). Thus, *Bill Clinton* would be represented by his home page.

Figure 3.1 shows the different possible combinations of these two dimensions. The first quadrant (SPI) depicts approaches (e.g., [31]) that use instance-level data from a single page to derive features. The second quadrant (SPP) depicts approaches (e.g,. [26]) that use full text from a single page. The third quadrant (MPI) depicts approaches (e.g., [106]) that use instance-level data from multiple pages. Fourth quadrant (MPP) methods (e.g., [101]) use full text from multiple pages to derive features.

Note that instance based approaches (SPI and MPI) can only be applied to entity topics and not topics in general as it would be challenging to find complex topics explicitly represented by specific phrases. Also approaches based on SPP and SPI data utilize information in only a single web page. The limitations of variations in the number of web pages and the level of data considered motivate us to formally study alternative methods for topic representation on the Web. Our own inclination is in exploring methods that utilize data from multiple web pages and beyond the instance level. This is because while a single web page may contain information relevant to a topic, it is unlikely to contain all the relevant information. Topical information is likely scattered across multiple pages, each potentially addressing different relevant aspects. Moreover, it is quite possible for relevant sentences to appear distant from sentences that contain an instance of the topic. E.g., in our dataset for experiment 1 a document relevant to the topic 'Hurricane Andrew' has the sentence *Hurricane*

Andrew was the most destructive United States hurricane of record, which is relevant and contains an instance of the topic. But four sentences away there is: *The vast majority of the damage in Florida was due to the winds.* This sentence is also relevant but does not contain an explicit instance of the topic.

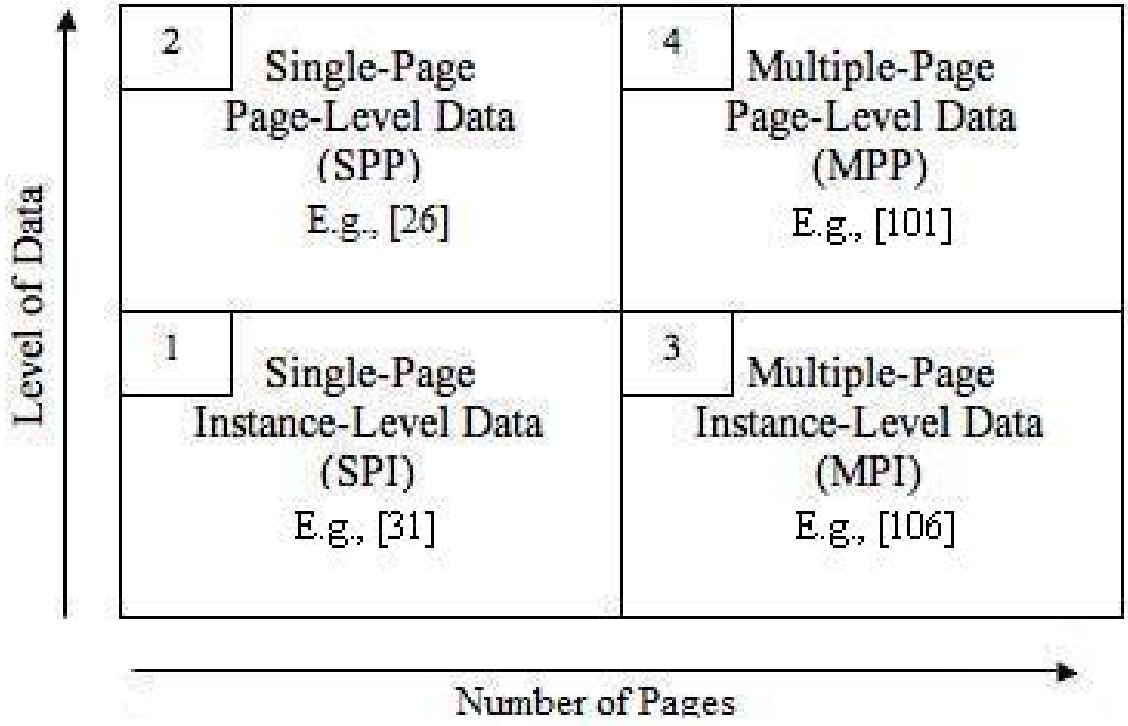


Figure 3.1: Different combinations of number of pages and level of data used, to generate representations. Number of web pages varies along horizontal axis and level of data varies across vertical axis.

Looking beyond representations, we find few knowledge discovery efforts [50, 106, 101] for exploring relationships between entities using information derived from multiple web pages. Some of these efforts are inherently limited because they utilize only small windows of text surrounding individual instances to represent an entity [106]. This means that other potentially useful information present in relevant web pages is ignored. Again such discovery efforts cannot be applied to general topics such as *Rainfall in the United States* because it is possible that relevant pages do not

contain this phrase.

The choice of features represents another significant dimension that differentiates various web mining approaches. E.g., in [101] the authors use words and named-entities while in [26] representations consist of words, links and subscribed mailing lists. Although in this thesis we concentrate on word, stem, entity, phrase, and link features, our long-term interest lies in building extensible representations from the Web that can accommodate various kinds of features. Additionally, prior efforts primarily use bags-of-words to represent entities. This leads to a lack of flexibility in the representations. A topic, especially representing a person entity, can have many different characteristics and consequently different relationships between topics can be established based on which characteristic is being considered. For example, a person may have much in common with another person based on the kind of work she does but may not have anything in common with the same person based on her personal interests. Prior efforts do not allow such differences to be considered. In our research, we seek a generic framework for representing topics using different kinds of features and allows for exploration of various relationships between them.

Another dimension that differentiates various web mining approaches is the method used to assign weights to features. Most efforts utilize either heuristics or probability distributions to assign weights to features. E.g., Adamic and Adar [26] assign $\frac{1}{log(tf)}$ weights to features, while Raghavan et al. [106] and Newman et al. [101] use probabilistic weights. Feature weights are important especially as the feature space is typically very large. We observe that this aspect has not been studied sufficiently in the context of knowledge discovery. Thus in this thesis we are interested in exploring different weighting methods and contributing new methods.

As mentioned previously, biomedicine has been a fertile domain for text mining research. However, most methods in this area have been designed specifically for data

from specialized sources such as MEDLINE. Recent studies have shown the benefit of using web data for tasks such as classification of biomedical documents [51], automatic recognition of entities in biomedical documents [58], and providing a baseline for comparison with specialized NLP models [85]. Inspired by such studies and our own background work in this area [111, 121, 120, 113], we are also interested in studying the feasibility of biomedical knowledge discovery on the Web and contributing methods for the same. In particular, in this thesis we explore the problem of predicting protein interactions using web data (chapter 7).

Finally, our research also addresses a fundamental mismatch in the level of information that the user typically desires and the level of information to which the user has access. For instance, when a user executes a query on a search engine, it retrieves a list of relevant web pages. A user would then have to further analyze the retrieved pages for relevant information. Users more likely want topic-level and not page-level information. This is illustrated in figure 3.2. Our long-term goal is to explore user interfaces that offer topic-level exploration rather than page-level explorations typical at present. Also, we believe that for web mining purposes topic-level information is more effective for analysis than instance or page-level information. As shown in figure 3.2, links connecting topics are possible. In fact a major goal in this thesis is to explore such links.

To conclude, we see that many opportunities for mining the Web remain untapped, offering a significant incentive for research in this area. Research with the goal of mining information from the Web is still at an early stage. We believe that this area will need aggressive research involving strategies from information retrieval, machine learning, and natural language processing.

Our overall goal is to do a systematic study and contribute new methods for topic representations and for knowledge discovery from the Web. Motivated by our

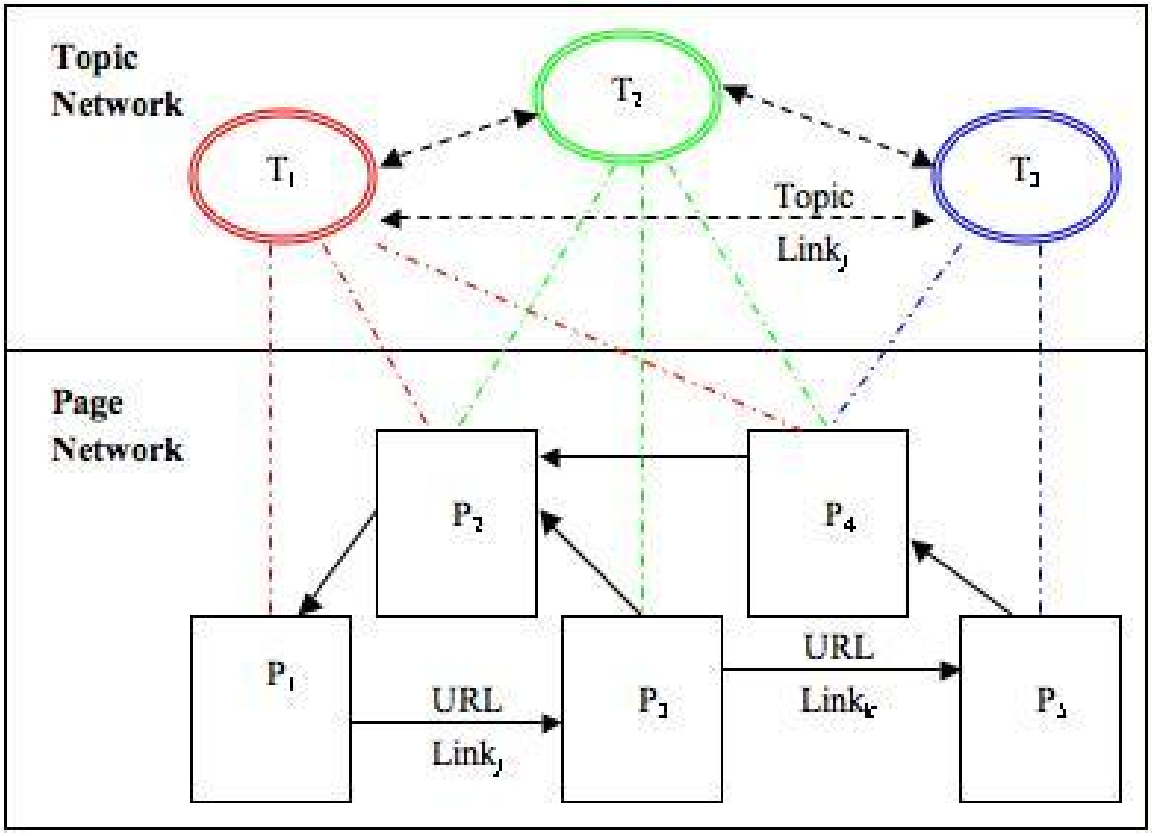


Figure 3.2: Illustration of Topical Web. Topics are represented by information scattered across multiple web pages compared to entities which represented by instance or single page data.

observations, we will study knowledge discovery on the Web keeping a topic perspective. Specifically, we propose a generic framework, of topic profiles, for representing topics on the Web. A profile consists of key features that characterize a topic and is derived from information in multiple web pages. Profiles enable analyzing a subject of interest at a level higher than instance or page-level. Profiles also provide the flexibility to deal with different aspects related to a topic. In the next chapter we describe topic profiles and in subsequent chapters describe applications of our profile framework for exploring relationships between topics using web data.

CHAPTER 4
TOPIC PROFILES

As mentioned in the previous chapter, we are interested in an approach for knowledge discovery from the Web that can handle different kinds of topics. Operationally, a topic implicitly underlies any search that a user employs and can be represented by the set of retrieved relevant pages.

4.1 Definition

A *topic profile* is analogous to a synopsis of a topic created from information present in relevant web documents. A profile identifies important features that characterize a topic. To illustrate, suppose a user is interested in the topic A, where A is a person. A's profile would consist of important features such as name, height, weight, address, field of interest, etc., important people and other entities A is connected to, such as company, spouse's name, etc., as well as the hyperlinks present in and pointing to A's documents. Figure 4.1 shows a hypothetical example profile for the topic *Bill Clinton*.

Defining a profile as consisting of 'important' features deliberately accommodates flexibility in feature selection. Depending upon the goal, certain varieties of features may be selected over others. This allows different aspects of a topic to be explored. For the purpose of this thesis we construct topic profiles using five types of features, viz., words, stems, noun phrases, named entities, and hyperlinks in retrieved documents.

Although we limit ourselves to these features in this thesis, the definition of a topic profile is extensible and can accommodate any type of feature. Also, each feature in a profile is assigned a weight, which highlights its relative importance to the topic at hand. This allows for filtering features by weight to retain those that are

Topic: Bill Clinton
Query: Bill Clinton OR William Jefferson Clinton
Number of Retrieved Documents: 10

Profile: (Top 3 features show for each type shown below)

Type	Term	Frequency	Num Docs	Weight	Rank
words	clinton	447	10	0.7964	1
words	president	144	10	0.2566	2
words	bill	103	8	0.1698	3
stems	clinton	537	10	0.7689	1
stems	presid	209	10	0.2993	2
stems	the	161	10	0.2305	3
phrases	bill clinton	11	8	0.2525	1
phrases	hope	9	9	0.2154	2
phrases	arkansas	9	9	0.2154	3
entities	clinton	281	7	0.6035	1
entities	bill clinton	204	9	0.4798	2
entities	william jefferson clinton	172	10	0.4193	3
links	http://en.wikipedia.org/wiki/Bill_Clinton#_note-First_In_His_Class	16	1	0.3068	1
links	http://en.wikipedia.org/wiki/1996	9	1	0.1726	2
links	http://en.wikipedia.org/wiki/Bill_Clinton#	7	1	0.1342	3

Figure 4.1: Example profile for topic *Bill Clinton*. Profile shows the top 3 word, stem, phrase, named entity and hyperlink features. Also shown are term frequency, document frequency, weight and rank.

important within the context of a given web mining problem.

In formal terms, a profile for topic T_i is defined as a composite vector consisting of one or more sub-vectors, having the following form:

$$Profile(T_i) = \{\{w_{i,1}f_{i,1}, w_{i,2}f_{i,2}, \ldots, w_{i,m}f_{i,m}\}, \{w_{j,1}f_{j,1}, w_{j,2}f_{j,2}, \ldots, w_{j,n}f_{j,n}\}, \ldots\} \quad (4.1)$$

Each sub-vector is composed of features of a particular type (e.g., words, entities, etc.). $f_{x,y}$ represents the y^{th} feature of type x (as mentioned above, in this thesis x can be words, stems, entities, phrases, and hyperlinks) and $w_{x,y}$ is the corresponding weight. Weights for individual features can be assigned using different methods. We describe the methods we explore later (chapter 5).

Topic profiles are different from other automatically generated structures that are similar, such as summaries or abstracts. The basic unit of a profile is a feature,

which can be of different types, while a summary or abstract consists of sentences. Individual features in a profile need not be cohesive while in a summary or abstract the sentences must "gel" together so as to form a cohesive unit. The emphasis in topic profiles is not necessarily human readability while a summary is primarily judged by its ability to present information that is easily comprehensible by humans. Our topic profiles are designed to support knowledge discovery, while, to the best of our knowledge, summaries or abstracts neither support nor have ever been used for this purpose.

The notion of a topic profile, as we define it, is motivated by the work done by Srinivasan on profiling biomedical topics using MEDLINE records [119]. Each MEDLINE record consists of some controlled vocabulary terms known as MeSH (Medical Subject Headings) terms. There are approximately 22,000 terms in MeSH. About 10 or so of these are manually assigned to each record by trained indexers at the National Library of Medicine (NLM). The MeSH term vocabulary is organized into semantic groupings known as *Semantic Types*. There are 134 semantic types and each MeSH term is assigned at least one semantic type. However, a MeSH term may fall under multiple semantic types. Srinivasan creates MeSH-based profiles from MEDLINE records retrieved for the topic. Each MeSH term is assigned a weight computed using the standard tf*idf formulation. Profiles may be limited to specific semantic types, which means that only those MeSH terms that come under the semantic types of interest are part of the profile.

4.2 Related Research

Interestingly, the literature reveals efforts that utilize structures that are somewhat similar to our profiles but used for different purposes. In [86], Li et al. create *entity profiles* limited to people. They use two types of features, salient concepts such

as name, organization, age, etc., and relationships to other entities (people). Glance et al. [65] generate a high level summary for a given product with features that specify 4 metrics, using data from blogs and message boards. These metrics are 'Buzz Count', 'Polarity', 'Author Dispersion' and 'Board Dispersion'. Using these metrics, they say that companies can gauge the opinions of consumers about particular products and improve their marketing intelligence. Liu et al. [88] attempt to discover topic specific information on the Web. Their goal is to identify the associated subtopics (aka salient concepts) from retrieved web pages (for the topic) and corresponding urls. This is analogous to a topic profile consisting of only 'subtopic' features. In [80], Kim et al. describe the Artequakt project. Their aim is to automatically create tailored biographies of artists using information present in web pages. Biographies are similar to topic profiles in which features are the important sentences related to a person. Factual features (along with sentences and paragraphs) and relations are extracted using IE (Information Extraction) tools. These are then filtered through an Artist ontology to eliminate non-relevant references. The filtered data is then combined to generate biographies. The extracted knowledge is also used to automatically update the ontology. Adamic & Adar [26] predict relationships between online instances of people via the similarity in certain characteristics mentioned on their home pages, such as hyperlinks, text, and subscribed mailing lists. Each of these can be thought of as features in a person profile. People with similar characteristics are predicted to be related.

While there are certain similarities between the research efforts described above and our topic profiles, there are also substantial differences. In general the goal of all these efforts is to create topical synopses based on select features. The type of features used varies across different efforts. The techniques described by (Li et al. and Adamic & Adar) and Glance et al. apply only to certain types of topics, viz., people

and consumer products, respectively. Their synopses are also limited to certain types of features. This limitation also applies to the other research efforts by Liu et al. and Kim et al. In contrast our research offers a general framework for profiling topics of any kind. Also, our profiles are extensible so as to accommodate new features and flexible to allow different combinations of features.

Another substantial difference lies in the potential applications of topic profiles. All of the efforts described above, except for (Adamic & Adar)'s, do not go beyond creating topic synopses, i.e., they do not consider mining functions. The one exception, (Adamic & Adar)'s work, is limited to using the information explicitly describing a person on a single web page. One of the major goals of this thesis to explore the issue of mining implicit or hidden information at the topic level. This is supported by our topic profiles.

4.3 Profile View

So far we have outlined five distinct types of features that can be included in a topic profile. However, a key property of topic profiles is flexibility. A profile can be created from any combination of features. We call this the *view* of a profile. For example, profiles can be built using only phrases and named entities or only outlinks in retrieved pages. A *view* specifies the particular aspects of a topic that are of interest.

Flexibility in the definition of a topic profile provides a general framework that supports different web mining methods. E.g., if profiles are built using only retrieved links and outlinks, then the process of mining relationships between the topics using such profiles is similar to some of the standard approaches that fall under web structure mining (e.g., [60, 84]).

4.4 Extending Profiles

Another key property of topic profiles is extensibility. They can easily be extended to accommodate features other than the ones we use in this thesis. These newer features can then be used to establish other kinds of connections between topics.

4.5 Profile Similarity

Profiles are formally defined as feature vectors. Thus, similarity between two profiles is defined as the similarity of their vectors. We can compute the similarity between two profile vectors using the cosine similarity measure from IR. Since our profiles are composite vectors, we extend the standard cosine formula. To compute the similarity between two profiles containing m feature types each, we compute pairwise cosine similarity between corresponding sub-vectors and take the average of the m sub-vector cosine scores. Formally, it is defined as:

$$Cosine(Profile(T_a), Profile(T_b)) = \frac{1}{m} * \sum_{i=1}^{m} \frac{\sum_{r=1}^{m_n}(w_{air} * w_{bir})}{\sqrt{\sum_{r=1}^{m_n} w_{air}^2 * \sum_{r=1}^{m_n} w_{bir}^2}} \quad (4.2)$$

where w_{air} is the weight of term r of feature type i in T_a's profile, w_{bir} is the weight of term r of feature type i in T_b's profile, and there are n features in the m^{th} feature sub-vector.

There are other methods we may use to combine sub-vector scores to determine profile similarity. E.g., in place of the average, we could compute the l^2-*norm* (see page 50 for definition) of the vector consisting of cosine scores of corresponding sub-vectors. A lower norm score would mean less overlap between two profiles vectors. We could also use a weighted average of sub-vector cosine scores, which would allow us to assign more importance to certain types of features, to determine profile similarity. We plan to investigate such alternate methods in future research. We choose the

simplest method in this thesis.

A similarity score of 0 means that the two profiles are completely dissimilar while a score of 1 means that the two profiles are completely similar. Note that profile similarity can be computed for different combinations of feature types. This allows for multiple types of relationships to be explored between two topics. Also, note that the cosine similarity does not depend on co-occurrence and consequently a relationship always exists between a pair of topics, unless the cosine score is 0. One can of course ignore edges below a threshold weight. We postulate a relationship between two topics if their profiles are 'sufficiently' similar. This relationship would be considered novel if the topics do not have any documents in common, i.e., they have not been previously explicitly connected (e.g., [125, 127]).

To conclude, in this chapter we have defined a profile-based approach for representing topics using information from web documents. A profile consists of different types of features, which characterize different aspects of a topic. Different combinations of features provide for different ways to view a topic. Profiles may be explored individually. They also provide the underlying framework on which higher level mining applications, such as mining relationships between topics, can be done. In the next chapter we describe the process for building profiles from web documents and our web-based implementation.

CHAPTER 5
METHODOLOGY AND IMPLEMENTATION

In this chapter we describe the process for creating topic profiles from web documents. We also present WebKD, a web-based implementation of our approach.

5.1 Building Topic Profiles

The process for building profiles consists of six steps. The input is a search query representing the topic and the final output is the topic profile. The output of each intermediate step is input to the next step forming a pipeline. This is illustrated in figure 5.1. We describe each step in more detail below.

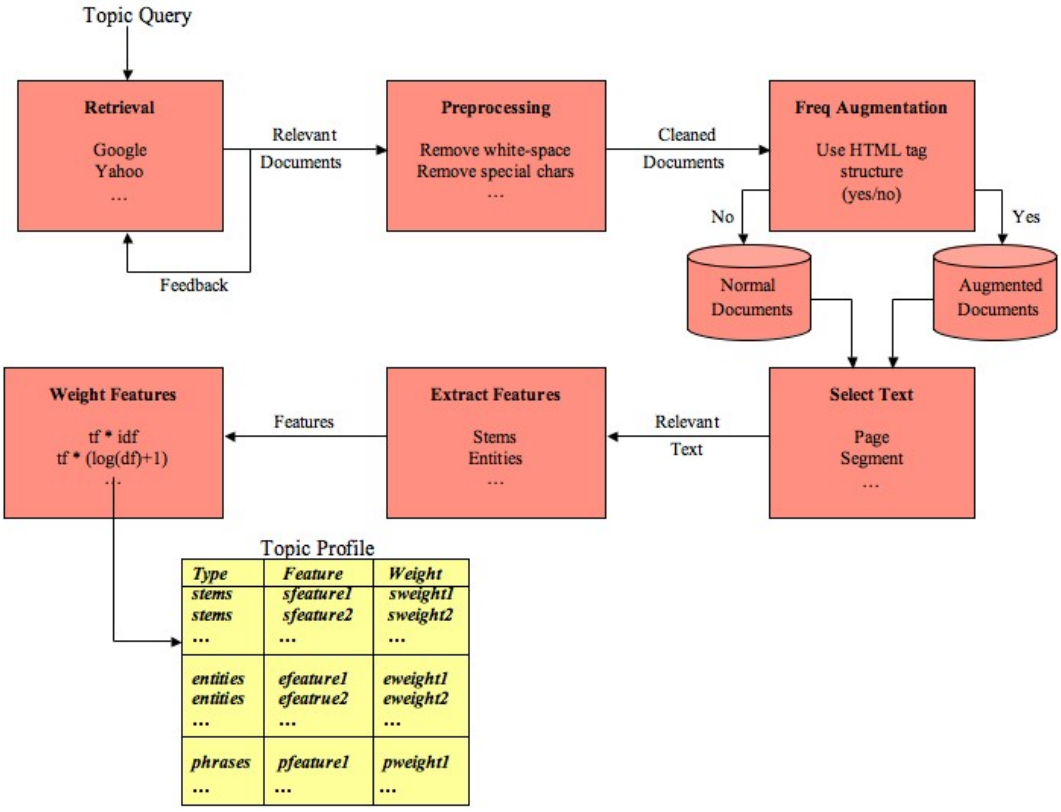


Figure 5.1: Pipeline process for building topic profiles. The input to the process is a topic query and the output is its profile.

5.1.1 Step 1: Retrieve relevant documents

Given a topic of interest, represented by a search query, the first step in building its profile is to identify documents related to the topic. This can be done by submitting the query to any of the major search engines such as Google, Yahoo! and MSN Search. The advanced search technology and filtering techniques implemented in these search engines ensure that unless the topic is ambiguous, at least the top ranked documents retrieved are mostly relevant.

Ambiguity in topic queries is a hurdle in retrieving relevant documents. Many terms are shared across domains and consequently reduce retrieval quality. E.g., the term *TOE* refers to a human body part and also is an acronym for the *Theory of Everything* in Physics. Named entities are particularly susceptible to this problem. E.g., the name *John Smith* is quite common and will retrieve documents referring to many different people.

Filtering to retain only relevant web documents is non-trivial and an active area of research (e.g., [69], [30]). While it is important to have the correct documents for building accurate profiles, it is not the central aspect of this research. Therefore, rather than spending a significant amount of time developing our own methods, we use existing solutions.

We use the Google search engine to fetch relevant web pages. However, our web-based implementation also provides access to the Yahoo search engine. Though it is beyond the scope of this thesis, it would be interesting to study the differences in profiles generated using the two search engines.

We retrieve the top N documents for a topic query where N is a parameter that can be empirically determined. In addition to the Web, topic profiles can also be generated from documents in a static corpus downloaded from the Web (or created in the same way). In this case the corpus can be indexed and relevant documents

retrieved using an off-the-shelf search engine, such as Lucene [2]. Generally speaking, we can further improve accuracy of retrieval using pseudo-relevance feedback, i.e., expanding the query using the most important terms from top ranked documents. This is an approach we use in chapter 9.

A separate but related question is what kind of search to use to represent an entity topic and a more general topic? We represent entities such as *Bill Clinton* and *Tom Cruise* using phrase queries. Thus, a page is considered only if it contains the entity phrase in the text. As it is very difficult to represent complex topics such as *Breast cancer in the United States in women between the ages of 35 and 50* using phrases, we use word queries to represent such topics. Here, only pages with all the words in the query are considered. More comprehensive queries can be created by combining topic synonyms using boolean OR operators. In that case documents that match requirements for any synonym are considered relevant.

5.1.2 Step 2: Preprocess retrieved documents

After relevant web pages have been identified, the next step is to preprocess them. This involves compacting the HTML source, i.e., removing extra whitespace, and removing non-ascii characters. Additionally we add sentence delimiters at various locations in the document. This is because web pages are created for a primarily visual environment. Thus, there is a major emphasis on tables, pictures and different typefaces and fonts. Because this information is displayed visually there is less need for common cues such as commas and periods. However, most current text processing tools require these cues to differentiate between different elements such as sentences and paragraphs. Table 5.1 lists all the changes we make to retrieved documents.

1. Remove blank lines from source
2. Remove non ascii characters from HTML source
3. Add sentence delimiter at
   - end of title
   - end of paragraph
     - </p>,

   - end of newline delimiter
     -
, </ br>
   - end of headline tag
     - <h1>, <h2>, <h3>, <h4>, <h5>, <h6>
   - end of row in table
     - </tr>
   - end of list item
     - , </dd>

Table 5.1: Steps for preprocessing retrieved web pages.

5.1.3 Step 3: Augment term frequency using tags

HTML is a semi-structured markup language. So unlike plain text documents, HTML provides markup tags that have different semantics. For example, text can be emphasized by surrounding it with certain tags tags such as <title> and <h1>. Such text are likely more important than plain text information present in the document. Other tags allow for grouping text in structures such as tables.

In prior research we find many studies that take into account HTML tags to improve retrieval. For example, Culter et al. [55] partition HTML tags into different classes and heuristically assign weights to each class. The frequency of text within tags in a certain class is augmented using the corresponding weight. Figuerola et al. [57] use a combination of text features, tag features in html pages, and information

Tag Class	**Weight**
Strong <b>, <i>, <u>, ,	8
H1-H2 <h1>, <h2>	6
Title <title>	4
H3-H6 <h3>, <h4>, <h5>, <h6>	1
Anchor <a>	1
Plain text	1

Table 5.2: Tag-based term frequency augmentation.

in backlinks to improve retrieval of Spanish documents. On the Web, early search engines utilized tag information in web pages for assigning a relevance score. For example, Alvatista used these clues as its primary scoring function. Currently, the major search engines, Google, Yahoo, MSN, etc., also utilize tag information for improving retrieval results. For example, when one does a search on Google, pages with query terms in the title are generally ranked higher. Tag information has also been used to enhance detection of *hubs* and *authorities* for a given topic [45]. In other application areas, tag-based weighting strategies have been used to improve document classification [37].

As mentioned previously, one of our goals in this research is to use the tag structure of web documents to improve the quality of our profiles. Thus, we use the tags to identify text that should be given more weightage. We do this by augmenting the frequency of the words. Following the work of Cutler et. al. we partition different

HTML tags into six classes. We assign a weight to each class and the frequency of words within tags of a certain class is augmented by the corresponding weight. Table 5.2 shows the weights assigned to different tags. Note that this procedure is not an exact science but is rather heuristic.

5.1.4 Step 4: Extract relevant text from documents

As mentioned in chapter 3 (see figure 3.1), prior approaches in creating topic representations have mostly focused on using either information from the full page or at the instance level. Most of these approaches (e.g., [106]) also focus on specific topics, primarily people entities. Instance level representations are derived by identifying individual instances (typically a phrase) of the entity and then using the information within a window of words surrounding the instance to derive its representation. We feel that both approaches are limited in certain aspects. Representations derived from the full page might contain non-relevant information while instance level representations might miss relevant information outside the windows surrounding individual instances. Also, in the latter case, it is not straightforward to decide the size of the window. Thus, we believe it worthwhile to also consider blocks of text that lie between full page and instance levels. In this research we consider four levels of data, viz., Full Page, Segment, Paragraph, and Sentence.

Full page level profiles are created from all of the text present in relevant web pages. Segment level profiles are created from relevant segments extracted from web pages. Paragraph level profiles are created from relevant paragraphs extracted from web pages. Finally, sentence level profiles, which are analogous to instance representations mentioned above, are derived from relevant sentences extracted from web pages.

In each case relevance is defined as the presence of the topic in the text. Entity

topics are identified using phrase-based search while more general topics are identified using word-based search. E.g., consider *Tom Cruise*. The entity topic Tom Cruise is denoted by the phrase query "Tom Cruise". Thus, in this case a page, segment, paragraph or sentence would be relevant only if it contains the phrase "Tom Cruise". On the other hand, the topic *Breast cancer in the United States* is denoted by the boolean word query 'Breast AND Cancer AND United AND States'. Here document relevance is judged by the presence of a majority of the terms in the query anywhere in the text and in any order. The same criteria applies to segments, paragraphs, and sentences. We do not consider stop words (such as 'a' and 'the') when judging relevance. The phrase vs. word distinction is important as there might be blocks of text that don't have the phrase "Breast cancer in the United States" but are still relevant. If synonyms are available then a sentence, paragraph or segment is considered relevant if it fulfills the above criteria for any of the synonyms.

In the examples above, we used the same query to retrieve documents and further identify relevant text blocks within the documents. But we can use a different query containing additional terms for the second task. Again considering the documents retrieved for the phrase query "Tom Cruise", we can be reasonably certain that any instance of 'Cruise' alone would in fact refer to Tom Cruise. Thus the query "'Tom Cruise" OR "Cruise"' would be more comprehensive and consequently better suited for extracting relevant blocks. However this query would not be good for document retrieval because among other things it would also retrieve documents for other people with the last name Cruise or for pages on Caribbean Cruises.

We use the MxTerminator software [108] by Ratnaparkhi to extract sentences from documents. MxTerminator uses a maximum entropy based approach to identify sentence boundaries in text documents. The installation package comes with a default model trained on Wall Street Journal text. While it is possible to re-train the model,

this requires a manually tagged corpus of sufficient size. We do not have access to such a corpus and creating one would be very challenging. Moreover in our research we do not limit ourselves to certain types of pages and training a model to accurately work with all kinds of web text further increases the challenge. Because of these reasons, we use the default model. We realize that this may not provide the most accurate results but for practical considerations this is our best option. Note that in chapter 6 we offer some error analysis.

In the English language a paragraph is a self-contained and semantically cohesive unit of text. By semantically cohesive we mean referring to the same point or idea. In web documents, paragraphs are delimited by or

 tags. Using these tags results in a blank line being inserted between paragraphs. We extract paragraphs from a document by splitting the text along blank lines. Since tables, etc. are also separated by blank lines from surrounding text, each table is considered to be a distinct paragraph. However, other tags such as <h1> or <h2> also result in blank lines being inserted between the headline text and the following text. Since a headline and immediately following text generally refer to the same idea, we consider them as jointly representing a single paragraph.

As described previously, relevance is judged by the presence of the entity topic phrase or general topic words within a block of text. For example, a paragraph would be considered relevant if it contains an phrase instance of an entity. However, some relevant paragraphs in a document may only contain an instance of the entity while others may only contain pronouns that refer to the entity. We would like to identify such paragraphs as relevant as well.

More generally, given an entity topic we would like to extract all segments of text that contain relevant information. Note that segments can be considered to be at a higher level than paragraphs and can span multiple paragraphs. Like a paragraph,

all the text in a segment refers to the same point or idea. Such a problem falls under the field of topic segmentation, which is an active area of research. Various methods for identifying topic segments in text can be found in the literature. A widely cited paper in this area has been written by Hearst [71]. In this paper she describes a segmentation algorithm, known as TextTiling. It is based on the hypothesis that the frequency of words varies across different topics. In other words a word that occurs very frequently in the context of one topic may not occur at all or occur with low frequency for another topic. Hearst divides a document into *pseudosentences* of equal length, i.e., same number of words, and combines psuedosentences to form *blocks* of text. Words that are most indicative of a block are discovered by converting blocks to term frequency vectors. Blocks are compared using the cosine similarity measure, which ranges between 0 and 1. In order to compare different sections of a document, the algorithm serially compares two overlapping blocks, advancing one pseudosentence at a time. This results in a set of cosine scores for the document, which are then analyzed for troughs. Troughs depict locations in the document where the similarity between blocks is very low and thus would constitute breaks in topics. Using a trough analysis algorithm, the system determines the best n troughs to consider as topic breaks. Suggested topic breaks are then moved to nearest real paragraph breaks. A C implementation of TextTiling created by Hearst is available online[1]. We use that implementation in this research. We use the default values for pseudosentence size (20 words) and block size (6 pseudosentences) and use blank lines as paragraph delimiters.

Using the TextTiling approach we divide a document into segments. Furthermore we mark segments as relevant and non-relevant based on whether they contain an instance of the entity. A weakness of the TextTiling approach is that it constructs segments only from adjacent paragraphs. This means that two paragraphs that are

[1]http://elib.cs.berkeley.edu/src/texttiles/

non-adjacent but are on the same entity would not be part of the same segment. If the latter does not contain a direct mention of the entity but has an indirect reference (such as a pronoun) then it would still be considered as non-relevant. To rectify such a situation we further process the relevant and non-relevant segments using an approach inspired by Chakrabarti et al.'s [45] work on identifying relevant and non-relevant pages in the context of a topic distillation problem. We first generate the centroid vector from all the relevant segments and for each relevant segment compute its cosine similarity with this centroid vector. We then compute the similarity of each initially deemed non-relevant segment and mark segments that have a similarity score greater than the median score for relevant segments, as relevant. Our segment-level profiles are formed from a combination of all relevant segments. This is detailed later.

5.1.5 Step 5: Extract features from text

The next step in the process consists of extracting features from the text. In this thesis we consider five types of features, viz., words, stems, phrases, named entities and links. We describe the process for extracting each type of feature in detail below.

Words

Words are one of the basic units of the English language and are typically separated from other words by a space character. We use a stoplist of common English words, such as 'a', 'the', etc., which are not very useful, and create profiles from only those words that are not in the stoplist.

Stems

A stem is a part of a word that is common to all its inflected and morphological variants. Inflection is the process of modifying a lexeme (smallest part of word that has

semantics) to make it grammatically consistent. E.g., consider the words 'attended' and 'attending'. Both have 'attend' in common, which is the stem. The 'end' and 'ing' are added to the stem to denote usage in past and present tense respectively. In IR stems are generally preferred to words because different variants are reduced to their root, which saves space and enables better searching. There exist many algorithms to automatically extract stems form text. The Porter stemmer by Martin Porter [105] has been widely used for stemming English language text. Lingua is a suite of various PERL text processing tools contributed by different people. We use the implementation of the Porter stemmer [62] contained in this collection. We first filter out all words in a document that are present in the stoplist mentioned above and use the stems of the remaining words in our profiles.

Phrases

Nouns and noun phrases are called "substantive words/phrases" in computational linguistics and "content words/phrases" in information science. They are used by many searching and indexing algorithms. Nouns and noun phrases can be extracted from both semi-structured and unstructured documents. Nouns can be easily recognized by using any of the available part-of-speech tagging applications such as the Brill tagger [38]. Typically, applications that identify noun phrases are built on top of taggers and employ pattern matching or inference rules. For example, the pattern $(Determiner) + Noun$, which corresponds to a determiner followed by a noun allows for extraction of noun phrases such as 'the whistle' or 'a bicycle'. Word and part-of-speech tag combinations that match such patterns are identified as noun phrases. In this research we use the part-of-speech tagging and noun phrase identification tool [61] that is part of the PERL Lingua suite. We limit noun phrases to a maximum of five words and extract only maximal noun phrases. Unlike with word and stem

features, here we do not use a stoplist to filter words. This is because stopwords are important for noun phrase identification.

Named Entities

Named entities constitute elements in the text that fall within predefined categories such as People, Companies, Locations, Products, etc. Named entities provide important information with respect to a topic. For example, the named entities in a documents relevant to a person A will likely illustrate the different people A associates with and places A visits. Various methods for named entity extraction are described in the literature and many off-the-shelf systems are also available. Most methods are based on either Linguistic grammars or statistical methods (Hidden Markov Models or Conditional Random Fields). We evaluated two off-the-shelf systems, viz., the Stanford Named Entity Recognition System (Stanford NER) [59] and ClearForest Tags [4]. Both systems utilize statistical models. We found that the ClearForest tool is more accurate and comprehensive than Stanford NER and consequently use the former in this research.

ClearForest [3] is a leading text mining company and develops various NLP tools such as a part-of-speech tagger, named entity tagger, web page content analyzer, etc. Some of these tools, including the named entity tagger are freely available via web-based services. We use the named entity tagger web service in this thesis. In terms of implementation, we use a SOAP client to request service from the ClearForest server. We send the text to be tagged via a SOAP request and the server sends back the tagged text in XML format. For each entity in the text, the server assigns it a class (from a predefined set of classes) and also specifies its position in the text. The XML also contains normalized versions of the entities. E.g., consider the text "Bill Clinton was the President. He lived in the White House.". Here the system recognizes both

'Bill Clinton' and 'He' as people entities and normalizes them to 'Bill Clinton'.

The ClearForest named entity tagger extracts only general entities and cannot be used extract entities from a specialized domain such as biomedicine. Recall that one of our goals is to use our profile-based approach for biomedical knowledge discovery using web data (see page 25). One of our experiments involves creating protein profiles. Therefore, we use a specialized named entity tagger to extract biomedical entities. Specifically, we use the named entity tagger that is part of the LingPipe suite of tools [14]. LingPipe is a suite of Java libraries and can be used for various types of linguistics analyses of text data. It contains statistically trained models for general and biomedical entity extraction. We are particularly interested in the latter. This model is trained on GENIA data [79] and can be used to extract biomedical entities such as genes and proteins.

Links

The URLs related to the retrieved pages constitute an important part of a topic profile. Link information for a topic can provide important clues about the neighborhood the topic is a part of. E.g., when profiling a person entity topic, the links from and to the retrieved documents may point to home pages of other people the person is related to, thus giving an indication of his/her social network. For example, some of the links (not shown in the figure) in the profile for *Bill Clinton* (figure 4.1) point to Wikipedia pages (somewhat similar to home pages) of prior Presidents as well as the home page of the Clinton Foundation.

Obtaining outlink information is relatively straightforward. Outlinks are mentioned in the documents themselves. Inlinks are also an attractive feature that can be added to profiles. The number of inlinks is a commonly used measure used to judge the importance of URLs. It is all the more attractive due to its correlation

with PageRank [39]. Inlinks can obtained by querying a search engine. However, due to strict limitations on the number of queries served per day by Google and Yahoo, we do not consider inlinks in this research.

5.1.6 Step 6: Assign weights to features

The final step in the process consists of assigning weights to the extracted features and normalizing the weights. Weights denote the relative importance of features within the context of a feature type.

In IR $tf * idf$ [109] is a standard method to weight terms. It combines the term frequency and the inverse document frequency. The inverse document frequency is defined as $log\frac{N}{df}+1$ where df is the number of documents in which a term occurs. The intuition behind this method is that terms that occur frequently within a document D but less frequently across a collection are good discriminators of D and should be assigned a higher weight in D's term vector. Additionally, language models assign probabilistic weights to terms from a probability distribution [104]. Weights are discounted to a certain extent using a smoothing function to allow for some probability mass to be reserved for unknown terms (e.g., terms in the query but not in any document). This provides the framework for computing conditional probabilities between a query and documents in the collection ($P(Q|M_d)$). Documents are ranked based the estimated probability that their corresponding language model (M_d) will generate terms in the query (Q). Unlike $tf * idf$, probabilistic weights are based only on the term frequency and not the document frequency.

In this research, we implement the standard $tf * idf$ weighting method. Additionally we implement several other methods that we believe are better suited for profiles. These methods are based on our own understanding and intuition of how term frequency alone or in combination with document frequency may be used to

weight features. These are listed below.

$$f_{wt1} = tf * idf = tf * [log\frac{N}{df} + 1] \quad (5.1)$$

$$f_{wt2} = tf * [log(df) + 1] \quad (5.2)$$

$$f_{wt3} = tf * \frac{1}{idf} = tf * [\frac{1}{[log\frac{N}{df} + 1}] \quad (5.3)$$

$$f_{wt4} = [\frac{wt.\ avg.\ tf}{avg.\ tf}] * idf = [\frac{wt.\ avg.\ tf}{avg.\ tf}] * [log\frac{N}{df} + 1] \quad (5.4)$$

$$f_{wt5} = [\frac{wt.\ avg.\ tf}{avg.\ tf}] * [log(df) + 1] \quad (5.5)$$

$$f_{wt6} = [\frac{wt.\ avg.\ tf}{avg.\ tf}] * \frac{1}{idf} = [\frac{wt.\ avg.\ tf}{avg.\ tf}] * [\frac{1}{[log\frac{N}{df} + 1}] \quad (5.6)$$

$$f_{wt7} = h\text{-}index_t \quad (5.7)$$

$$f_{wt8} = [\frac{wt.\ avg.\ tf}{h\text{-}index_t}] * \frac{1}{idf} = [\frac{wt.\ avg.\ tf}{h\text{-}index_t}] * [\frac{1}{[log\frac{N}{df} + 1}] \quad (5.8)$$

Equation 5.1 describes the $tf * idf$ weighting method, discussed above. In equations 5.2 - 5.8 we define our own term weighting functions based on different factors. The reasoning behind equations 5.2 and 5.3 is as follows. A profile is a description of a topic and contains features extracted from all retrieved documents. Unlike document retrieval where the stated goal is to identify features that best describe individual documents, here our goal is to identify features that best describe the topic. Given a collection of on-topic documents we believe the best features may be those that occur frequently within documents as well as across the document collection. Different documents provide different contexts and features that consistently occur in various contexts would intuitively be good at characterizing a topic. Thus, rather than penalize terms that occur in multiple documents as is done in document retrieval (via the *idf* factor) we assign higher weight to features that occur frequently across the document collection. In equation 5.2 the term weight is directly proportional to the number of documents it occurs in. In equation 5.3 the weight is inversely proportional

to the inverse document frequency, which translates to the higher weight being assigned to terms that occur in more documents. Here the overall size of the collection also plays a role.

In equation 5.4 we extend $tf*idf$ by using a weighted average for term frequency instead of simple term frequency. The weights are calculated based on the rank of the document (in the retrieved set) in which the term occurs. The intuition behind this is that terms that occur more frequently in top ranked documents should be considered more important than terms that occur more frequently in bottom ranked documents. The raw term frequency does not allow taking such differences into consideration. Thus, we take a weighted average where weights are inversely proportional to the corresponding ranks of the documents in which a term occurs. The weighted average is defined as:

$$Wt.\ Avg_t = \frac{\sum_{i=1}^{df}(1/r_i) * tf_i}{\sum_{i=1}^{df}(1/r_i)} \tag{5.9}$$

Here the sum is computed across all documents in which a term occurs, r_i is the rank of a document and tf_i is the frequency of the term t within that document. This factor is normalized by the average frequency to augment the weight of terms that occur less frequently. They represent rare features and may be important, especially in high ranked documents. As a side effect this method assigns low weight to terms that occur frequently in lower ranked documents. Similarly equation 5.5 is an extension of equation 5.2 and equation 5.6 is an extension of equation 5.3.

In equation 5.7 we use the h-index function to assign weights to terms. H-index [73] is a recently proposed method in citation analysis to evaluate the scientific productivity of an individual. A person has an h-index of h if he/she has authored h papers with each having at least h citations. It is a symmetric function and takes a more balanced view of productivity compared to the total or average number of

citations, which are other standard methods. The latter point is important because an individual will have a high h-index only if he/she has authored a large number of "important" papers rather than a few highly cited papers. In spirit, term weighting is a similar problem where the goal is to assign weights to terms based on their contribution within the context of the document collection relevant to the topic. Thus we modify the definition of h-index and use it for term weighting. A term t has an h-index of h if it occurs at least h times in h retrieved documents. For example, a term with an h-index of 2 occurs at least twice in 2 documents. This method assigns higher weight to terms that occur frequently in many documents. As stated previously, we believe such terms may best describe a topic. Equation 5.8 is a modified version of equation 5.6 with the h-index of the term used for normalization instead of the average frequency.

Finally, after weights have been computed, we normalize them using the l^2 vector norm. Each feature sub-vector is normalized independently. The l^2-norm is defined as follows:

$$l^2\text{-}norm = \sqrt{\sum_{i=1}^{n} w_i^2} \tag{5.10}$$

where n is the number of features in a feature sub-vector and w_i is the weight of a feature. This normalization step results in all sub-vectors having the same length, which is required to compute cosine similarity between two profile vectors.

5.2 WebKD: A Web-based Implementation

We have implemented our topic profiles approach as a web-based application known as WebKD[2]. WebKD allows individual users to create accounts and input topics to create profiles. Users can input a single topic or multiple topics at the same time. Users must specify values for various parameters including source of relevant

[2]Available at `http://lakshmi.info-science.uiowa.edu/WebKD/`

pages, number of pages, query type, data level and term weighting method. Users can choose between Google and Yahoo as the source of relevant pages and can also provide urls or documents in the input file (for multiple topics) to use to create profiles. Users can also provide a list of domains to be excluded when web pages are retrieved from the Web (using either Google or Yahoo). Queries can either be phrase-based for entity topics or word-based for general topics. The system offers page-level, segment-level, paragraph-level, and sentence-level profiles. Users must also specify the types of features to extract. As above, five types of features are supported, viz., words, stems, noun phrases, named entities, and hyperlinks. The system implements all of the term weighting methods described above and additionally implements weights based on term frequency alone and probabilistic weights. The system can create profiles for both general topics as well as biomedical topics. The major difference between the two is in the named entity features. For the former, we use the ClearForest tagging system while for the latter we use LingPipe. Appendix A contains some screenshot images of the system interface.

In terms of design, the system can be divided into two parts, viz., the front-end and the back-end. The front-end interface consists of Perl/CGI scripts while the back-end consists of a PostgreSQL database. Figure A.6 in Appendix A shows the back-end database schema. Users must create an account on the system before using it and all user information is stored in the database. To process a topic a user must set up a job for the system to execute. All information regarding the job, including the job name, topic(s), query(ies), retrieved documents and topic profile(s) are stored in the database. The system also keeps a history of all jobs that it has processed for a user. Users can at anytime view profiles for topics already processed. This design is inspired by Manjal[3] [112], a web-based biomedical text mining that we have created as part of another project.

[3]Available online at http://sulu.info-science.uiowa.edu/Manjal.html

To conclude, in this chapter we have described the process for building topic profiles from web documents and presented an overview of the system implementation. Our procedure uses several off-the-shelf text processing tools. We have also designed several term weighting strategies tailored to web profiles. In the next chapter we present an experiment to evaluate different profiles generated using the various options mentioned above. We also analyze different kinds of errors in the profile building process. In chapter 7 we evaluate different profiles based on their accuracy in predicting relationships between topics, specifically protein topics. In chapter 8 we explore social networks of US senators using profiles and compare with networks generated from US Senate voting records. In chapter 9 we present an application of profiles for expert search.

CHAPTER 6
EXPERIMENT 1: ASSESSING QUALITY OF INFORMATION: COMPARING WITH WIKI PROFILES

In the previous chapter we described the individual steps for building topic profiles. Furthermore, we presented different options for various parameters such as term weights, level of data used, types of features, etc. Our first goal is to assess the quality of our web-based profiles. We do this in two ways. First we assess the quality of information using Wikipedia as a gold standard in this chapter and second we test these profiles in terms of predicting known relationships between topics in the next chapter.

6.1 Objective

In this experiment our goal is to assess topic profiles by the extent to which the information contained is relevant. Essentially we create profiles using web data for a variety of topics. We evaluate them using Wikipedia, a known high quality source of information. Since Wikipedia articles are structured differently compared to our profiles, we first build profiles from the appropriate Wikipedia entries and then compare these with our web-based profiles.

6.2 Gold Standard Data

Wikipedia [16] is an online repository of information on various topics in different languages. The English version of the site contains descriptions of more than two million topics[1]. A Wikipedia entry for a topic typically contains a summary of the topic, a small table containing a list of prominent characteristics, a list of relevant external links, and a list of references. A Wikipedia entry can be created by anyone and can also be edited by anyone. Thus, well developed entries tend to contain the

[1] as of October 2007

viewpoints of many people. Wikipedia is the largest collaborative journalism effort till date and has come to be viewed as a highly regarded reference site [87]. It has been reported that the quality of Wikipedia articles is high and they are used by many teachers as a reference [18]. Wikipedia articles are also frequently cited by newspapers [137]. There are also prior research efforts in text mining where Wikipedia has been used as a high-quality resource, such as to enhance background knowledge for text categorization [64] and for Question-Answering [41]. In addition to Wikipedia, another source of gold standard information is the Encyclopaedia Brittanica (EB). In prior research [114] we compared profiles generated from Wikipedia and EB and found interesting differences. For example, the latter contained only cursory information for many of the topics in our set and was comprehensive for only a few popular topics such as *World War II*, while the former contained much more information for a broader range of topics.

Figure 6.1 shows the Wikipedia entry for the topic *Bill Clinton*. A major difference between a Wikipedia entry and our profile structure is that the former contains an English language summary of a topic while the latter instead contains a list of key words/phrases associated with a topic extracted from different sources on the Web. Also Wikipedia entries are manually created and edited whereas our web profiles are automatically created from a combination of sources retrieved from the Web.

6.3 Topic Set

For our evaluation, we compiled a set of fifty topics belonging to three categories, viz., companies, celebrities and events. 21 companies were randomly selected from the Fortune 500 list for 2006 and 16 celebrities were randomly selected from the Forbes celebrity list for the same year. We randomly compiled the 'events' topics from an online source containing a list of 30 major 20^{th} century events[2]. Examples

[2]http://history1900s.about.com/cs/majorevents/

Figure 6.1: Wikipedia record for *Bill Clinton*.

topics include *Tom Cruise*, *WWII*, and *Dell Corporation*. For each topic we identified corresponding Wikipedia pages and created gold standard profiles. We then compared profiles built from web data with profiles built from the corresponding Wikipedia pages.

For each topic we manually identified synonyms and created a boolean (OR) query. We used phrase-based queries as they are particularly suited for representing entities such as names of people, companies, events, etc. An example search is *"World War II" OR "WWII"*. We then retrieved the top 100 web pages for each query using the Google search engine. The retrieved sets were filtered to exclude pages from approximately 600 web sites known to mirror Wikipedia content[3] (including Wikipedia itself). The choice of using the top 100 retrieved pages is somewhat arbitrary and is

[3]http://en.wikipedia.org/wiki/Wikipedia:Mirrors and forks

primarily motivated by the daily limit on the number of queries imposed by Google. Also, in prior research [114] we looked at the effect of the number of pages and found that the quality of a profile improves only up to a certain threshold (50-100 pages) after which there isn't much change. After retrieving the documents we created different kinds of profiles for each topic. We evaluate each kind of of web profile by its average similarity with gold standard profiles as per equation 4.2. This allows us to compare different methods for building profiles.

6.4 Experimental Design

A key challenge in evaluating these profiles is that we are exploring a wide variety of profile building methods spanning multiple dimensions, with multiple options within each dimension. The dimensions include: term weighting strategies, the kinds of features extracted, the amount of information used on a given page, and the use of tag information. Clearly considering all combinations of options across all the dimensions would make our experiment and analysis of the results unmanageable. Hence we proceed systematically through the evaluation exploring one dimension at a time. Each time we pick the best performers and move to the next dimension. Thus we first explore term weighting strategies, then the use of tag information in web pages, followed by the amount of information to use from a page. Across each of these dimensions, we build profiles with different kinds of features.

6.5 Exploring Different Term Weights

Our first goal is to assess profiles built using the different term weighting methods listed in section 5.1.6. One of these methods ($tf * idf$) is commonly used in IR research while the others are based on our own understanding of the problem. Our initial focus is on comparing the term frequency component present in most weighting

methods. There are two standard methods for considering the frequency of a term. One is to use the raw term frequency, i.e., if a term t occurs 200 times then its term frequency is 200. Another standard method is to use the augmented term frequency. The augmented term frequency is defined as:

$$wt_{augm} = 0.5 \ + \ 0.5 \ * \ \frac{tf}{max\_tf} \tag{6.1}$$

where tf is the raw term frequency and max_tf is the maximum frequency of any term in the collection. Augmented term frequencies are always between 0.5 and 1 and boost the score for terms that occur less frequently in the collection relative to terms that occur more frequently. E.g., a term that occurs 10 times in a corpus with the frequency for any term being at most 50, will have an augmented term frequency of 0.6. Prior to evaluating the different term weighting methods we compare these two term frequency variants.

Figure 6.2 shows the average similarity (and 95% confidence intervals) of web profiles generated with $wt1$, $wt2$ and $wt3$ weights (defined in section 5.1.6) with normal (raw) and augmented term frequencies, with corresponding Wiki profiles. Here profiles contain stem features. We see that the average similarity for profiles with features weighted by raw term frequency is significantly higher than profiles with augmented term frequency weights. This implies that the former is better at assiging higher weights to relevant features than the latter. As an example, the average similarity is 61% higher for $wt2$ weighted profiles with a raw frequency component than $wt2$ profiles with an augmented frequency component. We believe this to be the case because augmented term frequency artificially raises the importance of low frequency terms, which we believe are not the best characteristics of a topic. While augmented term frequency has been shown to be better for document retrieval, generating profiles is a different kind of problem, where features must be representative of

a set of documents rather than a single document. Given these results, in subsequent experiments we use raw term frequency.

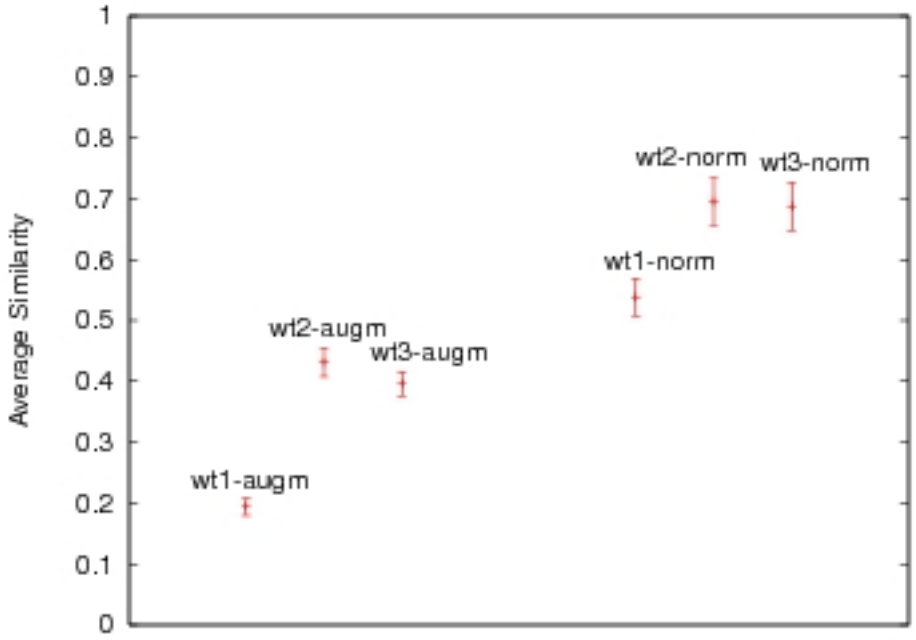


Figure 6.2: Average Similarity (with 95% confidence interval). Comparing term weights with raw (norm) and augmented (augm) term frequency components. Profiles contain stem features.

Now we compare the eight different weighting methods. Figures 6.3 and 6.4 show the average similarity of profiles with Wiki profiles for each term weighting method. In 6.3 profiles contain stem features while in 6.4 profiles contain stem, word, phrase, entity and hyperlink features.

From the figures the first conclusion we draw is that $tf * idf$ ($wt1$) performs poorly. This is interesting as it is a standard strategy in document retrieval. More generally, profiles where weights are directly proportional to the document frequency ($wt2$ and $wt5$) or inversely proportional to the inverse document frequency ($wt3$ and $wt6$) have the highest average similarity and are significantly better (at 0.05 level) than the profiles where weights are inversely proportional to the document frequency

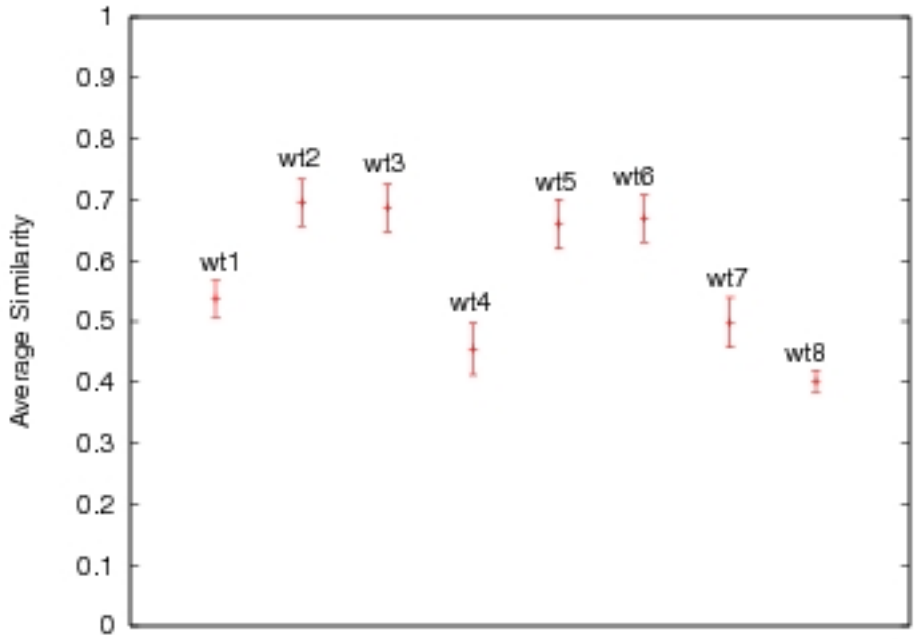


Figure 6.3: Average Similarity (with 95% confidence interval). Comparing different term weighting methods. Profiles contain stem features.

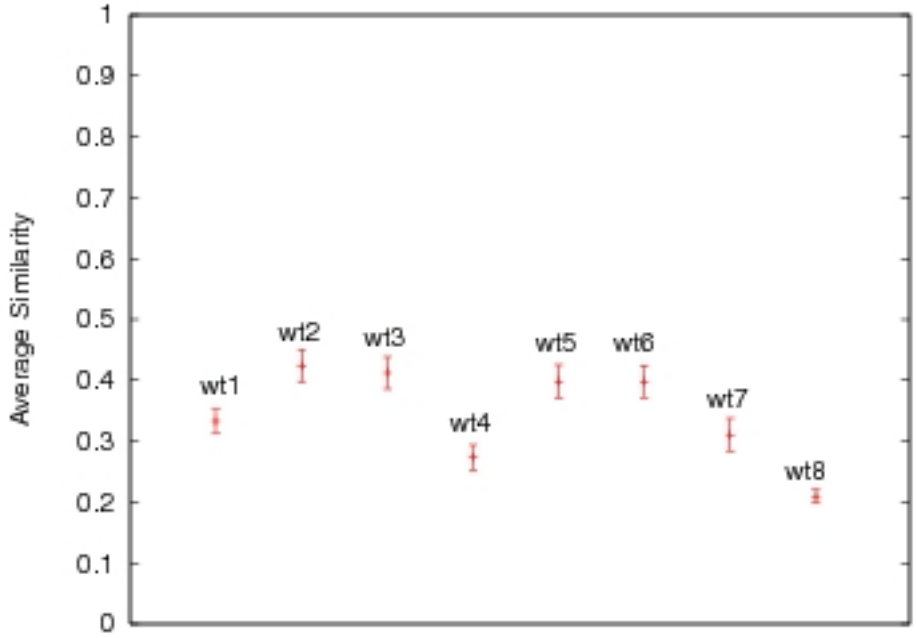


Figure 6.4: Average Similarity (with 95% confidence interval). Comparing different term weighting methods. Profiles contain word, stem, phrase, entity, and hyperlink features.

($wt1$, $wt4$). Next, comparing methods that have a weighted average term frequency component ($wt4$, $wt5$, $wt6$) with corresponding methods that use raw term frequency ($wt1$, $wt2$, $wt3$) reveals no significant differences. Somewhat surprisingly, the h-index weighting method ($wt7$) has fairly low average similarity. This is surprising because we believe that assessing the importance of terms is in many ways similar to assessing the importance of people and the h-index has been successfully applied to the latter problem. Our plan for future research is to probe this new weighting strategy further to see if we can get a better understanding of its performance. Finally we conclude that normalizing by the h-index ($wt8$) leads to a significant degradation in performance when compared to normalizing by the average term frequency ($wt6$).

Overall, the highest average similarity is for profiles built with $wt2$ weights. Recall that $wt2$ is a multiplicative combination of the term frequency and a factor directly proportional to the document frequency ($tf * [log(df) + 1]$). The average similarity for $wt2$ weights is however not significantly different than the average similarities for $wt3$, $wt5$, and $wt6$ weights. But since $wt2$ requires the least amount of computation, we prefer this strategy.

Also, we see that profiles with stems features alone have a much higher average similarity than profiles with multiple types of features. This could be a consequence of the different number of sub-vectors in the two cases. Observe that in the latter case there are five sub-vectors and the similarity is an average across these five. We also observe that most of the Wikipedia pages for the topics we chose do not have many hyperlinks, which leads to a fairly low similarity in terms of hyperlink features. For example, the average similarity between web profiles and Wiki profiles consisting of only $wt2$ weighted hyperlink features is 0.0255. This in-turn brings down the average. We also observe a similar downward trend for named entity features.

6.6 Exploring the Value of Tags

We next assess the impact of augmenting the frequency of terms based on the specific tags within which they occur. Recall that one of our goals in this thesis is to utilize the tag information present in semi-structured web (HTML) documents. The method by which we use tag information was described in the previous chapter (section 5.1.3). Figures 6.5 and 6.6 show the average similarity scores for profiles using tags to determine term frequency. Also shown are the average similarity scores for corresponding profiles where the tag information is not used. The first figure shows scores for profiles with only stem features and the second figure shows scores for profiles with multiple types of features. Scores are shown for the four best term weighting methods identified in the previous experiment (figures 6.3 & 6.4).

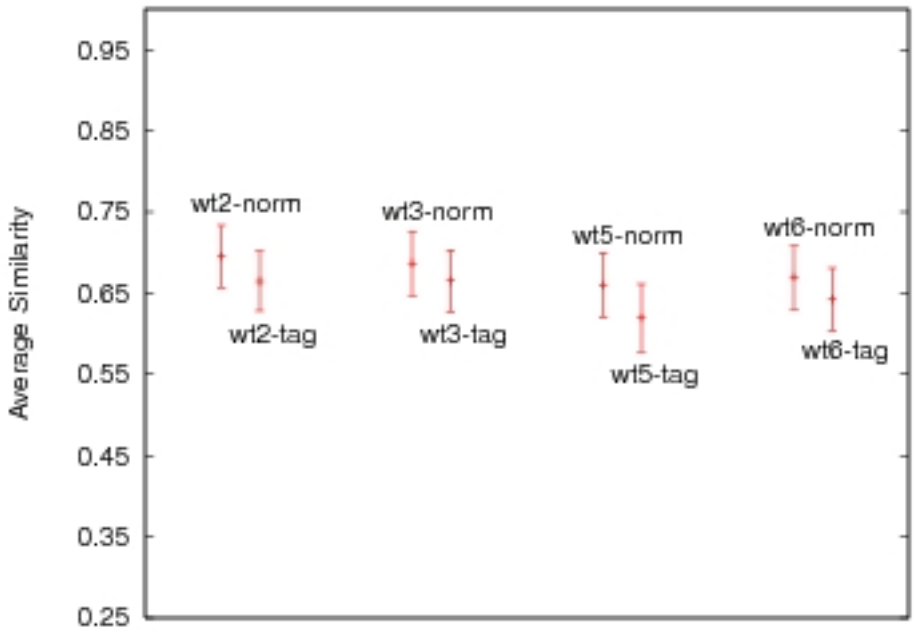


Figure 6.5: Average Similarity (with 95% confidence interval). Comparing tag augmented term weights with unaugmented weights (norm). Profiles contain stem features.

From figures 6.5 and 6.6 we conclude that using the tag information to augment

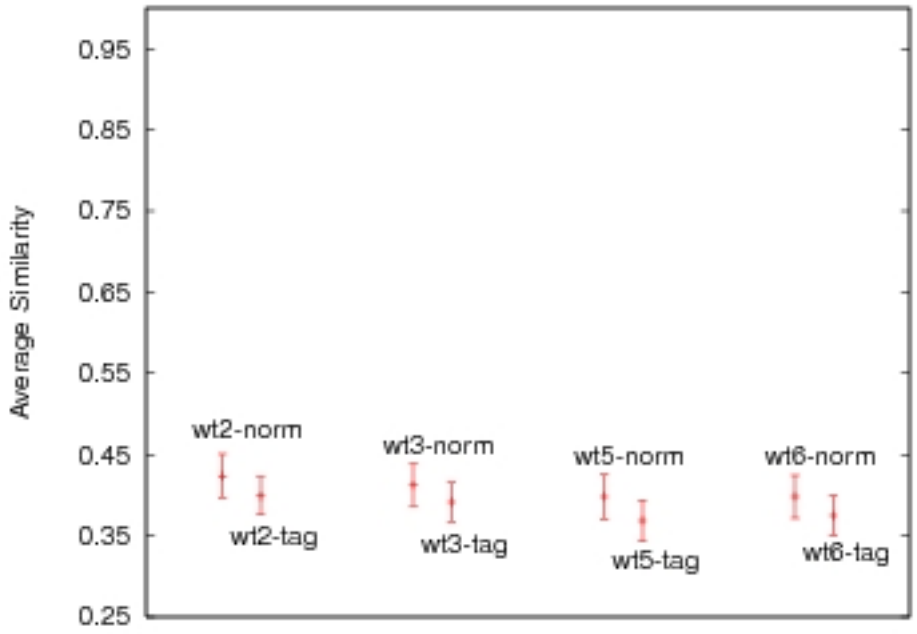


Figure 6.6: Average Similarity (with 95% confidence interval). Comparing tag augmented term weights (tag) with unaugmented weights (norm). Profiles contain words, stems, phrases, entities, and link features.

the term frequency results in a lower average similarity, although not significantly so, than ignoring the tag information. This is true for all four weighting methods, and for both stem only profiles and profiles with multiple types of features. In an attempt to understand why this is the case, we manually inspect the top features in some of the profiles with tag augmented term frequency as well as corresponding unaugmented frequency profiles. For example, table 6.1 shows the top 10 word features for the topic *Tom Cruise* for the two types of profiles.

As the table illustrates, we find that the top ranked features are mostly the same for both methods but the corresponding weights in the tag augmented profiles are lower. Using the tags is successful in boosting the weights of some relevant features, as in for the word 'actor' in the example above. But overall the weights of features, including relevant features, are lower due to the l^2-norm (refer to section 5.1.6 for definition) factor being higher. This consequently results in a lower similarity score.

Unaugmented Frequency			Tag Augmented Frequency		
word	**frequency**	**weight**	**word**	**frequency**	**weight**
cruise	1777	0.6455	cruise	4346	0.6033
tom	1558	0.5660	tom	3974	0.5517
news	460	0.1559	tv	2329	0.2773
tv	488	0.1520	episode	2710	0.2314
scientology	444	0.1436	news	1260	0.1631
katie	398	0.1262	scientology	1144	0.1409
episode	468	0.1045	katie	734	0.0889
movie	263	0.0869	top	652	0.0844
top	244	0.0827	movie	636	0.0803
holmes	233	0.0715	actor	575	0.0708

Table 6.1: Comparison of top 10 word features in unaugmented and tag augmented profiles for topic *Tom Cruise*.

However, this strategy may be useful in a different setting, such as where the rank of the features rather than the weight is more important. We also remind the reader that the particular set of multiplicative factors we chose to modify term frequency came from a paper by Culter et al. [55]. It may be that although they worked for document retrieval, these may not be appropriate for profile building.

6.7 Exploring Different Levels of Data

Finally, we compare profiles generated from various levels of data, viz., page, segment, paragraph and sentence, extracted from relevant web pages. We explore this aspect using the $wt2$, $wt3$, $wt5$, $wt6$ weighting methods, not augmented using tags. Figures 6.7 and 6.8 show the average similarity of different kinds of profiles with corresponding Wiki profiles.

Figure 6.7: Average Similarity (with 95% confidence interval). Comparing different levels of data. Profiles contain stem features.

The figures shows that page, segment and paragraph-level profiles are similar in their average similarity and sentence-level profiles have much lower average similarity for all the term weighting methods. The difference between page and sentence is significant at the 0.05 level for $wt2$, $wt3$, and $wt6$.

Comparing page, segment and paragraph -level profiles further, the first has on average 6843 stem features and 33952 features overall (stems, words, phrases, entities, and hyperlinks) while the second has 5609 stem features on average and 23304 features on average overall. As expected paragraph profiles have the lowest number of features among the three with 4239 stems features on average and 16577 features on average overall. This means that although segment and paragraph contain significantly less number of features than page they do not lose much relevant information. So in terms of space, segment and paragraph -level profiles are preferable (especially where

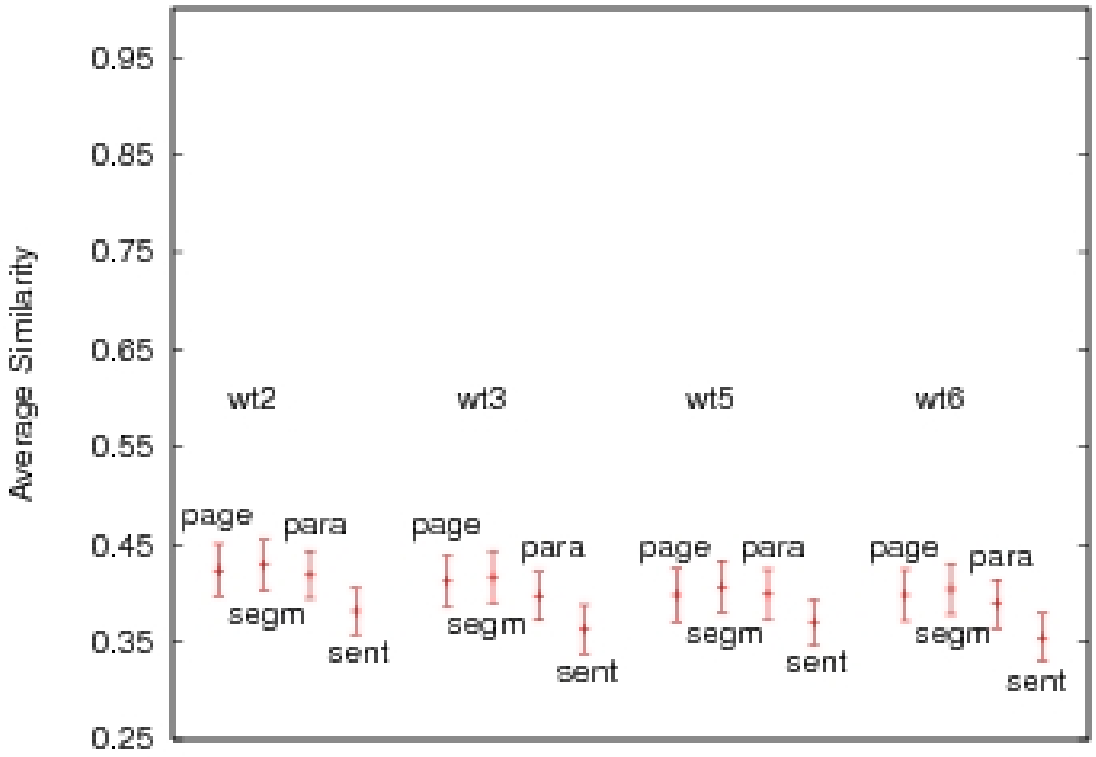


Figure 6.8: Average Similarity (with 95% confidence interval). Comparing different levels of data. Profiles contain words, stems, phrases, entities, and link features.

memory or storage are important issues). This also means that our segment extraction strategy successfully meets its primary goals. However, page-level profiles are the easiest and the fastest to generate, generally taking less than a 5^{th} of the time it takes to generate segment and paragraph -level profiles[4]. Thus when space is not an overriding issue or time is critical, page-level profiles may be a better choice.

To summarize, our best profiles consist of features weighted using $tf*[log(df)+1]$ weights, where the term frequency component is not augmented using tag information and features are extracted from the full-text in web pages.

[4]This is a rough estimate.

6.8 Error Analysis

Our approach for building topic profiles relies on several underlying technologies, such as document retrieval, sentence detection, etc. Each of these constitutes a potential source of error. In this section we look at different types of error and quantify them in terms of precision, recall and f-score. We create a small test dataset of 6 topics out of the 50 that we compiled for the experiment above. We randomly choose 2 companies, 2 celebrities and 2 events, listed in table 6.2. Then we manually analyze the types of errors listed in table 6.3. Our goal in manual analysis of errors is to further understand performance and obtain valuable information needed to improve upon our methods.

Topic
1. Chevron corporation
2. Home Depot
3. Nicole Kidman
4. Johnny Depp
5. World War II
6. Florida Keys Hurricane

Table 6.2: Topics for error analysis.

Type of Error
1. Retrieval error
2. Sentence detection error
3. Segment extraction error
4. Entity detection error
6. Phrase detection error

Table 6.3: Errors in profile building process.

We use precision, recall and f-score to to measure error. Precision is a standard measure of accuracy and is defined as

$$P = \frac{TP}{TP \ + \ FP} \tag{6.2}$$

where TP denotes the number of true positive and FP the number of false positive predictions. Recall is used to measure comprehensiveness and is defined as

$$R = \frac{TP}{TP \ + \ FN} \tag{6.3}$$

Here FN is the number of false negatives. The f-score measure is a combination (harmonic mean) of precision and recall. When equal importance is given to both it is defined as

$$F = \frac{2 \ * P \ * \ R}{P \ + \ R} \tag{6.4}$$

where P and R are precision and recall respectively.

6.8.1 Retrieval error

The first step in our profile building process is to retrieve relevant web pages using a web search engine (Google). In order to analyze the error in a broad cross-section of retrieved pages, for each topic we create a stratified sample of pages consisting of the top 5 retrieved pages and then 5 pages from the top 5 - 20 pages, 5 pages from the top 20 - 40 pages and so on. Our sample consists of 30 pages in total for each topic. We then manually judge them for relevance and calculate overall precision and precision in the top 5 retrieved documents. We use a generous criteria for relevance, specifically we consider a page to be relevant if it contains at least one sentence that conveys relevant information about the topic. Table 6.4 shows the precision scores and precision scores in the top 5 retrieved pages for each topic.

Topic	**TP**	**FP**	**Precision**	**Precision@5**
Chevron Corporation	30	0	1.0	1.0
Home Depot	27	3	0.9	1.0
Nicole Kidman	30	0	1.0	1.0
Johnny Depp	30	0	1.0	1.0
World War II	30	0	1.0	1.0
Florida Keys Hurricane	18	12	0.6	0.2

Table 6.4: Precision for all retrieved pages and top 5 retrieved pages.

We see that for most topics the search engine is very accurate in retrieving relevant pages. However, for the topics *Home Depot* and *Florida Keys Hurricane*, some non-relevant documents were retrieved. In the first case, 3 pages describing steps to take to save energy that were sponsored by Home Depot were retrieved. These pages do not contain any information about the company itself and thus are not relevant. In the second case, many of the retrieved pages describe the florida keys hurricane evacuation route which is not related to the Florida Keys Hurricane of 1935. Thus, the precision is fairly low for this topic. This may be because our query (the phrase "florida keys hurricane") may not have been specific enough.

6.8.2 Sentence detection error

The next type of error we analyze stems from the sentence detection tool that we use to identify sentences from web pages. Recall that we use the Mxterminator package that implements a maximum entropy model to detect sentence boundaries. We apply Mxterminator on the same stratified set of web pages that we created above and manually judge each sentence that it outputs. Table 6.5 shows the precision, recall and f-score for each topic.

We see that the precision, recall and f-scores are high for topics that would have

Topic	**TP**	**FP**	**FN**	**Precision**	**Recall**	**F-score**
Chevron Corporation	91	51	30	0.6408	0.7251	0.6803
Home Depot	190	27	16	0.8756	0.9283	0.9012
Nicole Kidman	450	673	460	0.4007	0.4905	0.4411
Johnny Depp	186	548	349	0.2534	0.3477	0.2932
World War II	479	52	42	0.9021	0.9194	0.9107
Florida Keys Hurricane	43	37	50	0.5375	0.4624	0.4971

Table 6.5: Precision, recall and f-score for sentence detection tool.

more structured and clean relevant web pages such as companies and events. Most of the pages retrieved for the former were official pages and were fairly well structured. In the latter case, because of the seriousness of the events the retrieved pages had a more somber tone and were generally created by experts in these fields. Therefore, these pages were also fairly well structured. Not surprisingly the scores are very low for celebrities. For these topics most of the retrieved pages were created by fans and generally did not follow any structure. In such a case, the algorithm, which is trained on a news corpus, which generally consists of well structured pages, is not a good fit. Additionally, in general since the algorithm is not trained on web data, it does not take into account certain unique features of web pages, such as headings, line breaks (
), tables, etc., and fails to correctly identify certain sentences.

6.8.3 Segment extraction error

The next type of error we analyze comes from our own method to identify relevant segments from web pages to create segment-level profiles. We applied our method to extract segments from the retrieved pages in the stratified sets above and manually judge them for relevance. Recall that a segment is relevant if it contains at least one instance of the topic. Table 6.6 below shows the precision, recall and f-score

for segments extracted by our method for each topic.

Topic	**TP**	**FP**	**FN**	**Precision**	**Recall**	**F-score**
Chevron Corporation	7	2	0	0.7778	1.0000	0.8750
Home Depot	7	18	0	0.2800	1.0000	0.4375
Nicole Kidman	42	8	3	0.8400	0.9333	0.8842
Johnny Depp	14	5	6	0.7368	0.7000	0.7179
World War II	10	12	26	0.4545	0.2778	0.3448
Florida Keys Hurricane	2	7	0	0.2222	1.0000	0.3636

Table 6.6: Precision, recall and f-score for segment extraction method.

We see that the precision is high for three of the topics, i.e., both celebrities and one company, and fairly low for the rest. For topics with low precision, we observe some of the errors stem from errors earlier in the process. E.g., precision here is lowest for *Florida Keys Hurricane*, which also has the lowest retrieval precision. Other reasons for error include the loss of structural (presentation) information when web documents are treated as text documents (after removing the tags). E.g., in a page retrieved for *Home Depot* one of the menu items contained a link to the Home Depot company page, for which the corresponding anchor text was 'Home Depot'. This menu item were merged with subsequent text, which did not contain relevant information, by the segmentation method. Because of the presence of the query terms, such a segment is considered as relevant. Recall errors were primarily a function of retrieved pages generally containing information on a broad set of relevant but independent sub-topics. For example, many of the pages retrieved for *World War II* contained descriptions of a number of separate events that happened during the war. In this case while some sub-topic segments contained the query terms, others did not and were thus considered non-relevant. Furthermore, due to the low similarity,

the latter segments do not cluster with the relevant segments.

6.8.4 Entity detection error

Next we analyze errors in extracting named entities from the text. As mentioned previously, we use the ClearForest named-entity tagger to identify general named entities. We use the tool to tag all the pages in the stratified set for each topic and then manually judge all the named-entities identified by the system. Table 6.7 below shows the precision score for each topic.

Topic	**TP**	**FP**	**Precision**
Chevron Corporation	90	28	0.7627
Home Depot	129	44	0.7457
Nicole Kidman	721	110	0.8676
Johnny Depp	491	30	0.9424
World War II	510	22	0.9586
Florida Keys Hurricane	58	13	0.8169

Table 6.7: Precision of named entity extraction tool.

The results show that the ClearForest system performs consistently well for all the topics. This is likely due to the fact that the it is trained to handle web data directly as opposed to other text processing tools (such as Mxterminator) that are designed for more well-formed non-web text documents. Most of the errors were due to missing contextual cues (e.g., 'gong li rachel mcadams rachel mcadams askmencom' was identified as one entity, being a row in a table). Unfortunately, we were unable to find a general pattern in the errors.

6.8.5 Phrase detection error

Finally, we look at errors in detecting noun phrase features from documents. As mentioned previously, we use the Lingua::EN::Tagger PERL package to identify noun phrases. We first apply this tool to the stratified set of retrieved pages for each topic. Then, we randomly select 25 noun phrases for each topic and manually evaluate them. Table 6.8 shows the precision score for each topic.

Topic	**TP**	**FP**	**Precision**
Chevron Corporation	16	9	0.64
Home Depot	19	6	0.76
Nicole Kidman	19	6	0.76
Johnny Depp	20	5	0.80
World War II	16	9	0.64
Florida Keys Hurricane	15	10	0.60

Table 6.8: Precision of noun phrase extraction tool.

As was the case with the named entity recognition system, we see fairly consistent precision across all the topics. This is likely because the Tagger package depends only upon part-of-speech cues to identify noun phrases, and in terms of the usage of the English language, generally there isn't any difference between web documents and other documents. Most of the errors are due to the lack of other contextual cues, such as commas and periods (e.g., 'tipoff. 17 december 2006 nicole') which the system uses as boundaries. Also, some errors are due to the nature of HTML documents where structural cues are more visible in nature and thus hard to represent in plain text. Thus, 'emma watson askmen.com premium' is recognized as a single noun phrase but 'emma watson' and 'askmen.com premium' are acutally present in two cells in a row in a table in the original HTML document.

To summarize, we analyzed various sources of error in building topic profiles. We observe that the most significant errors are in detecting sentences and segments while document retrieval has the least number of errors. Errors in document retrieval can be minimized by using more precise queries. We use MxTerminator to detect sentence boundaries in web pages. MxTerminator was trained on a corpus of Wall Street Journal articles, which have very different characteristics than web pages . Retraining MxTerminator on a more representative set of documents would help to reduce errors. We also observe that our approach for identifying relevant segments produces mixed results. Our immediate goal is to explore different strategies to improve recall and precision. One approach would be to consider similarity thresholds other than the median to decide which segments not containing query terms to consider as relevant (see section 5.1.4).

6.9 Discussion

In this chapter we evaluated different methods for building topic profiles based on the quality of information in the profiles for different kinds of topics. Our results show that page-level profiles containing stem features with $tf * [log(df) + 1]$ weights contain the most amount of relevant information. Interestingly the standard $tf * idf$ weighting strategy does not perform well. We also present a novel use of the h-index, for term weighting. Although it does not perform as well as our best method, the performance is comparable to $tf * idf$, which is encouraging and motivates future research. We have also presented a description of the various types of error and provided statistics to judge the extent of each error. We found that tools (Google, ClearForest) that have been designed with a web perspective consistently perform well while there are certain failure points for other tools (MxTerminator, TextTiling) that have been designed for cleaner and more well structured data.

CHAPTER 7
EXPERIMENT 2: PROFILES FOR PREDICTING PROTEIN INTERACTIONS

In the previous experiment we focussed directly on the profile building methods using Wiki as the gold standard. Here we test profiles in the context of a text mining application. Specifically we test their ability to predict known relationships between different proteins using their web profiles. Additionally, a secondary goal (refer chapter 3) is to determine the feasibility of using web data in a "bioinformatics" context.

7.1 Background and Related Research

Extracting protein interactions from text is a well-known problem in the biomedical domain. Various approaches for this problem can be found in the literature. Most focus on extracting interactions from MEDLINE records, specifically from the title and abstract. For example, Blaschke and Valencia [34] use simple pattern matching to identify sentences in MEDLINE abstracts that match predefined patterns based on co-occurrence of certain action verbs such as 'interacts', 'activates', etc, with protein names. An example pattern is "proteinA activates proteinB". They assume that protein names are given. Positive sentences are further analyzed to deal with negative information as well as contradictions. Unfortunately, they do not present an evaluation of their approach.

In [135] Thomas et al. describe a customized version (for biomedical text) of Highlight, a general-purpose Information Extraction system. From a small training set of MEDLINE abstracts they manually generate linguistic templates that in different ways describe interactions between proteins. They assign a score to each template reflecting its confidence. Their approach has a precision between 69% and 77% and recall between 29% and 55%.

Marcotte et al. [92] use a Bayesian probabilistic approach to identify MEDLINE abstracts that describe yeast protein interactions. They assign log-likelihood scores to abstracts based on the frequency distribution of 500 *discriminating* words, which they determine from a training set consisting of abstracts of 260 articles on yeast proteins in the DIP database. Example discriminating words include 'binds', 'interacts', and 'associates'. Testing their approach on 325 abstracts, they were able to correctly identify 77% of the relevant abstracts.

Park et al. [102] use significantly more comprehensive techniques than partial parsing to identify protein interactions in MEDLINE abstracts. Their approach consists of combining a part-of-speech tagger with rules to identify unknown words. They also utilize a regular grammar to propose noun phrases around a pre-compiled list of verbs describing interaction and a combinatory categorical grammar (CCG) to validate them. This approach yields a precision of 80% and recall of 48%.

Sugiyama et al. [123] compare four different machine learning based methods, viz., kNN, Decision Tree, Neural Network, and SVM, to automatically identify sentences describing protein interactions in MEDLINE abstracts. They utilize different features around the verb of a sentence, including the verbal form, part of speech information of words surrounding the verb, etc. Additionally, they use characteristics of a noun in the sentence, such as whether it contains numeric figures, non-alphabetic characters, or upper case letters. They tested their classifiers on 1000 MEDLINE articles and obtained precision in the range of 0.623 (kNN) to 0.881 (SVN) and recall in the range of 0.647 (kNN) to 0.881 (SVN).

In [47] Chen and Sharp describe Chilibot, a system for mining Pubmed abstracts. Chilibot uses a variety of NLP techniques to create rich descriptions of genes, proteins, drugs, and more general biomedical concepts. The authors describe applications for Chilibot in exploring relationship networks as well as novel hypotheses

discovery. They evaluated the system by using it to predict 770 known relationships between proteins specified in the DIP database and had a precision of 96% and recall ranging from 90.1% to 91.2% depending on the number of abstracts considered.

The Web is a rich source of information on various biomedical topics. Many of the widely used online biomedical databases such as Pubmed, SwissProt, and GeneCards are indexed by the major search engines. Consequently records from these databases are retrieved by biomedical web queries. As mentioned in chapter 3 recent studies have explored the benefit of using web data for various tasks such as classification [51] and named entity recognition [58]. To the best of our knowledge there are no prior research efforts that explore the use of web data for predicting relationships between proteins. This in part provides the motivation for this experiment.

7.2 Gold Standard Data

We use the Database of Interacting Proteins (DIP) [6] as our source for gold standard information. DIP is a manually curated database of experimentally determined protein interactions. As of November 21, 2006 the DIP database contains data on over 19000 proteins and 55000 interactions compiled from over 62000 experiments described in the published literature and other sources. Data from DIP has been used as a gold standard by a number of text mining research efforts in the biomedical domain [47, 35].

7.3 Experimental Design

We randomly select 82 human proteins from DIP. According to DIP 90 pairs of these proteins interact with each other. For each protein we identify relevant synonyms from the SwissPROT database [10]. To avoid ambiguity, we ignore synonyms that are also English words. We then create phrase-based boolean (OR) web queries

from the names and corresponding synonyms and retrieve the top 100 pages from Google for each query. No pages are retrieved for two of the protein queries resulting in our final gold standard set consisting of 80 proteins and 88 interactions.

For each protein we create various types of profiles differing primarily in terms of the level of data used (page, segment, paragraph and sentence) and the type of features. We use the $wt2$ term weighting method, which we previously established as our preferred method. Also based on the previous experiment we do not use tag information and we use the raw term frequency. Three types of features were extracted from the documents, viz., stems, noun phrases, and named entities. For named entity features we use LingPipe [14], which is a specialized biomedical extraction system, rather than the general named entity tagger (ClearForest) we used in the previous experiment.

As mentioned in section 4.5, we use the standard IR cosine similarity to measure the similarity between profiles. We predict a relationship between two proteins on the basis of their profile similarity. Two proteins are predicted to be related if their similarity score is above a certain threshold, set through training runs. Based on the predictions, we measure precision, recall and f-score defined in equations 6.2, 6.3, and 6.4, respectively. In this experiment we consider f-score as our primary measure to evaluate different types of profiles.

We adopt a 5-fold cross validation approach. The 80 proteins we selected yield 3240 unique pairs, which are randomly split into 5 equally sized groups. We consider the union of 4 of these splits as our training set and the remaining split as our test set. This allows us to iterate over the data 5 times resulting in a 5-fold cross validation design. We determine the optimum threshold for the training set by iterating over different similarity thresholds and select the one that maximizes the training f-score. We then apply this threshold to the test set and calculate the test f-score. We compare

the different types of profiles based on the average of their five test f-scores.

7.4 Results

Table 7.1 shows the average training and test precision, recall, training and test f-scores, and test f-score 95% confidence intervals for profiles built from different levels of data. Here profiles contain stem features.

	Training			**Test**		
Type	Prec	Rec	F-score	Prec	Rec	F-score (95% CI)
Page	0.1687	0.2782	0.2096	0.1498	0.2458	0.1830 (0.1096, 0.2564)
Segment	0.1941	0.3240	0.2427	0.1849	0.3114	0.2315 (0.1694, 0.2936)
Paragraph	0.1796	0.3040	0.2255	0.1681	0.2736	0.2076 (0.1576, 0.2576)
Sentence	0.0988	0.2328	0.1372	0.0808	0.1903	0.1126 (0.0763, 0.1489)

Table 7.1: Average training and test f-scores (with 95% confidence interval) for stem profiles derived from different levels of data.

The results show that segment-level and paragraph-level profiles have the highest average test f-scores. Both perform better, although not significantly so, than page-level profiles and are significantly better than sentence-level profiles. Page-level profiles utilize the full text present in documents. Although we consider only the top ranked documents retrieved by Google, this does not guarantee that all the documents are relevant. Furthermore, even within documents certain portions may not be relevant. In the previous experiment, where we measure the information content of profiles, this did not have a detrimental affect probably because we were comparing with Wikipedia profiles, which by the virtue of being generated from high quality (manually generated) data mostly have relevant features. However, in this experiment the similarities are being calculated between pairs of web topic profiles. Thus the non-relevant features common across the topics have a definite impact, especially

in terms of boosting the similarity of non-interacting pairs of proteins. This results in lower precision. Also, the f-scores for all four types of profiles are fairly consistent across the training and tests, which means that our method generalizes well. In this regard, segment-level profiles are the most consistent.

Table 7.2 shows the average training and test results for profiles with entity features. Here we see that paragraph-level profiles have by far the highest average test f-score and are significantly better than page and segment -level profiles. This is in contrast to the results we obtained with stem profiles (table 7.1). Again, the training and test f-scores are fairly consistent for the different types of profiles indicating that our methods generalize well. Here paragraph-level profiles are the most consistent. Comparing with the previous results, the average f-scores are higher for entity profiles than for stem profiles derived from sentence and paragraph -level data and lower for profiles derived from page and segment -level data. This may be because entities are generally much more specific and more selectively used than words (from which stems are derived), which means that they are better indicators of relevance. To summarize, our results suggest that entities that co-occur with the topic of interest in sentences or paragraphs are powerful features.

	Training			**Test**		
Type	Prec	Rec	F-score	Prec	Rec	F-score (95% CI)
Page	0.1663	0.2385	0.1943	0.1178	0.1986	0.1407 (0.0816, 0.1998)
Segment	0.1563	0.4229	0.2268	0.1386	0.3657	0.1974 (0.1789, 0.2159)
Paragraph	0.2157	0.4772	0.2970	0.2212	0.4764	0.2999 (0.2569, 0.3429)
Sentence	0.2026	0.3996	0.2654	0.1785	0.3560	0.2363 (0.2140, 0.2586)

Table 7.2: Average training and test f-scores (with 95% confidence interval) for entity profiles derived from different levels of data.

Table 7.3 shows the results for different types of profiles with noun phrase

features. Again we see that paragraph-level profiles have the highest average test f-score, followed this time by page-level profiles. However, none of the differences are statistically significant. Comparing with prior results we see that the average f-scores are much lower for profiles with phrase features than for profiles with stem or named entity features. One possible reason for this is that we use a general English noun phrase extraction tool that is not capable of identifying only the biomedical phrases. Given that we are creating profiles for proteins, biomedical phrases would probably be better features than general phrases. In the case of named entities we use a specialized biomedical named entity tagger. Finally, the training and test f-scores are again fairly consistent across the different types of profiles with the best case being page-level profiles.

	Training			**Test**		
Type	Prec	Rec	F-score	Prec	Rec	F-score (95% CI)
Page	0.1468	0.1901	0.1398	0.1237	0.1680	0.1398 (0.0912, 0.1884)
Segment	0.1316	0.1690	0.1479	0.0932	0.1286	0.1079 (0.0259, 0.1899)
Paragraph	0.1180	0.2731	0.1642	0.1080	0.2421	0.1489 (0.0725, 0.2253)
Sentence	0.1195	0.1395	0.1282	0.0929	0.1157	0.1022 (0.0501, 0.1543)

Table 7.3: Average training and test f-scores (with 95% confidence interval) for phrase profiles derived from different levels of data.

7.5 Discussion

Our results show that named entities are the best indicators for predicting protein relationships. Paragraph-level profiles with entity features have the best performance and are significantly better than most of the other types of profiles. These observations are in contrast to our observations from the previous experiment where

we found that page-level profiles with stem features contain the most relevant information. This leads us to conclude that the type of profile that should be created depends primarily upon the problem and that this choice can have a significant impact on the results.

In comparison with a more specialized biomedical system (Chilibot), our best method has both lower recall (0.48 vs. 0.91) and lower precision (0.22 vs. 0.96). This comparison is rough as it ignores differences in experimental design and dataset used. We believe a significant reason for lower precision and recall is that our approach is based purely on web data while Chilibot uses much more specialized and high-quality data from Pubmed. To understand this further, we plan to build profiles for the same set of proteins using MEDLINE data and compare those results with the results we obtain here with web data. Another avenue that we would like to explore in the future concerns feature selection. From the results we observed that our best profiles consist of biomedical entities, which are more relevant to this domain than general features such as stems and noun phrases. We believe that our profiles will greatly benefit from the presence of more domain-related information. The named entity tagger that we used in this experiment (LingPipe) comes pre-trained on the GENIA corpus [79], which is a collection of 2000 abstracts on molecular biology. Consequently LingPipe only extracts the specific types of named entities in GENIA. Clearly more biomedical features can be extracted from documents. In this respect, tools such as MetaMap [29] that identify different kinds of biomedical concepts in text may prove useful.

CHAPTER 8
EXPERIMENT 3: PROFILES FOR EXPLORING SENATOR NETWORKS

In this chapter we present exploratory research on profile-based social networks for US senators. We create these profiles from web data and compare them against social networks generated from more structured data, specifically from US Senate voting records.

8.1 Background

A principal focus in text mining research is the identification and analysis of relationships between different entities. Much research has been done in this regard within different application areas such as biomedicine [75, 36] and business intelligence [32]. In this context, Social Networks Analysis (SNA) provides a natural framework for studying different entities, especially people, and their relationships. A social network is a social structure of entities that are tied by one or more interdependency relationships. Formally, it can be represented by a graph where nodes represent entities and edges represent relationships between entities. Social networks are studied in many different research areas, such as geography, social science and information science and have been applied to understand, among other things, dissemination of information between people, informal interactions between companies via their employees, friendship networks, diffusion of innovations, and the spread of diseases.

The Web is a rich resource of information on people and thus offers an excellent opportunity for Social Network Analysis. Information about a person may be obtained in many ways. For example, information may be explicitly posted by the person, indirectly inferred from records of their patterns of activities (e.g., search logs) or may be derived from references in other sources, such as news reports and blogs. This is especially true for people who occupy public positions such as Senators

and Representatives of the US Congress.

We are motivated to study senators because their viewpoints and decisions affect the daily lives of millions of people. Consequently they are the subject of discussion on a large number of web pages. Also, senators have been the subject of prior research in computer science. E.g., Wang et al. [138] find groups of senators using descriptions of positive votes on certain issues. Raghavan et al. [106] use the influence of neighbors in a vote-based senator social network to categorize individual senators into different groups. Using a language models based approach, they create pseudo-document descriptions, which they call entity models, of entities using data from fixed windows surrounding individual instances of an entity in retrieved documents. They also use these representations to identify relationships between entities that may not be explicitly related in any document.

In Political Science, there is interest in the roll-call voting patterns for senators and countries in the US Senate and the United Nations General Assembly, respectively, for purposes other than standard statistical analysis [48]. Roll call votes have been explored to identify politically disadvantaged groups of people [68], study the unity and flexibility of US political parties compared to other countries [97], and to study the effect of roll-call voting patterns on senatorial elections [43]. Voting patterns may help gain a better understanding of the position of a senator on various issues, which in turn may lead to a better understanding of the workings of the Senate.

Many times one's choices may be at least partially influenced by one's characteristics and background. E.g., a senator from Texas is perhaps more likely be against gun control than a senator from California. We investigate this aspect in the context of the senator voting patterns. Specially, we are interested in exploring whether similarity in web profiles is mirrored by similarity in voting patterns. A question asked is: do senators with similar profiles vote alike and vice-versa? More generally, how

do profile-based social networks of senators created from web data compare against social networks created from roll-call voting data?

We wish to state here that although we use objective methods to assess individual graph properties and make comparisons, some of the interpretations are likely to be subjective. This is typical of research in social networks.

8.2 Related Research

There is significant research on automatic extraction and analysis of social networks from web data [78, 26, 106, 54, 91, 93], mainly for person entities. The various methods differ primarily on how individual entities are represented, and how relationships between entities are inferred. Moreover, most are limited to representing entities using either instance-level data [78, 93] or page-level data [26, 54]. These strategies were explained earlier in chapter 3. Most methods also infer relationships between entities based solely on their co-occurrence in one or more web pages [78, 54, 91, 93]. Few methods [106] use data from multiple pages to represent entities. Also, very few methods [26, 106] go beyond co-occurrence to infer relationships between entities. These are some of the advantages we offer in our general profile-based approach.

We believe that topic profiles can provide an effective means for building informative social networks. Also, we offer flexible methods to infer relationships between entities. Entities with very similar profiles are likely to be strongly related while conversely entities with very dissimilar profiles are likely to be weakly related or totally unrelated. Relationships between entities can also be characterized by common features in their profiles. Thus, we believe profile-based networks will allow us to overcome the limitations mentioned above and are well-suited for social network analysis.

8.3 Voting Data

We use roll-call voting data from the US Senate. We downloaded voting data from the US Senate website[1] for all Congressional sessions from 1989 onwards up to the second session in 2006. For each vote, we obtained the vote id, short title, description, date, and the result (whether it was passed or rejected). We also obtained the individual votes cast by each senator. In this research, we limit ourselves to votes cast by the 100 individuals who were senators in November, 2006. Overall these 100 senators have cast 6026 votes during the time period we consider.

Our first challenge with this data was to determine the topic of each vote. This is a non-trivial task that is challenging to automate. As an example, a vote titled *A bill to amend the Immigration and Nationality Act to change the level, and preference system for admission, of immigrants to the United States, and to provide for administrative naturalization, and for other purposes* is on immigration. Taking inspiration from Raghavan et al. [106] we are particularly interested in votes belonging to the topics of abortion, immigration, defense, economy, and education.

The categorization of votes was done manually by two students not involved in any of the other aspects of the research. We asked the two student judges to categorize (Yes or No) each vote under the five different categories mentioned above. Votes that were not under any category were categorized as 'other'. We randomly divided the votes in equal numbers among the two judges, ensuring that votes in a single Congressional session were not split. For each vote, judges based their decision primarily on the title and description available. A single vote could be placed under multiple categories. For example, vote S. 1160 in the 1^{st} session in the 101^{st} Congress in 1989, titled *Armstrong Amendment No. 324; Relative to Chinese nationals who flee coercive population control policies*, was judged as both abortion and immigration related. The 6026 votes were categorized as 82 abortion related, 139 immigration

[1] http://www.senate.gov

related, 730 defense related, 2179 economy related, and 380 education related.

We later decided to limit the exploratory research in this thesis to only abortion and immigration. Hence, votes in the remaining three categories were included in the 'other' category. We plan to explore the other topics in future research.

We calculate the Kappa statistic [49] to evaluate the agreement between the two judges. The Kappa statistic (equation 8.1) is a standard measure for assessing inter-rater agreement and the reliability of annotated data. We did this with 100 votes randomly picked from the 6026 votes. Each vote required three binary decisions for the three categories, resulting in a total of 300 decisions.

$$\kappa \;=\; \frac{P(Observed\ Agreement) \;-\; P(Chance\ Agreement)}{1 \;-\; P(Chance\ Agreement)} \tag{8.1}$$

	General		
	No	Yes	Total
No	239	5	244
Yes	22	34	56
Total	261	39	300
P(Observed Agreement)	0.9100		
P(Chance Agreement)	0.7320		
Kappa (κ)	0.6642		

Table 8.1: Kappa statistic to evaluate inter-annotator agreement.

In table 8.1 the rows show the number of Yes and No judgments for one judge and the columns show the numbers for the other judge. Overall the Kappa score is 0.6642, which means that the two judges are largely in agreement with each other.

8.4 Experimental Design

8.4.1 Web Profiles

For each senator we conduct a web search on Google using the full name and nickname. To reduce ambiguity we ensure that their retrieved pages contain the term 'senator'. An example query is 'senator AND (James DeMint OR Jim DeMint)'. For each senator we retrieve the top 100 pages and create profiles from the full-text. The names are searched as phrases.

In this experiment, profiles consist of stem and named entity features. We also compare with profiles created from paragraph-level data containing the same features. We do not use the tag information to augment frequency (see section 5.1.3 for tag-based profiles).

The profiles as described above are from a general search, i.e., not specific to any issue. However, quite often senators, even from the same party, agree on certain issues while disagreeing on others. Thus, we are also interested in analyzing issue-specific senator votes and profiles, specifically on abortion and immigration. We create issue-specific profiles by adding the issue to the general query for a senator mentioned above. For example, we use the query 'senator AND (James DeMint OR Jim DeMint) AND abortion' to retrieve abortion-related pages for Senator DeMint. Again profiles are created from the full-text of the top 100 retrieved pages.

8.4.2 Senator Networks

For each pair of senators, we calculate profile similarity (refer to section 4.5) and use this to estimate of the strength of their relationship. We create multiple social networks, each differentiated by the type of features in the profiles. Specifically, we create a network from profiles with stem features and a network from profiles with entity features. We term such networks as *Passive Networks* because the relationships

do not depend upon the actions of the actors in the network.

We also create a social network for the senators from the voting data. In this network, a senator is represented by his/her votes in the Senate. We use the Jaccard score to measure the strength of the relationship between two senators. The Jaccard score (equation 8.2) is a measure of the overlap in the voting records of two senators. In the equation, S_i is a vector ($\{v_{i1}, v_{i2}, \ldots, v_{im}\}$; i = 1,2) that represents the m votes[2] cast by the Senator on issues voted upon in the Senate between 1989 and 2006.

$$Jaccard(S_1, S_2) = \frac{S_1 \bigcap S_2}{S_1 \bigcup S_2} \tag{8.2}$$

A score of 1 indicates that the two senators have always voted the same while a score of 0 indicates that they have always voted differently. In contrast to the profile networks, we terms such networks as *Active Networks* because the relationships depend completely upon the actions of the actors in the network.

8.4.3 Filtered Networks

We also consider networks with thresholds applied to edge weights. This allows us to ignore relationships (similarities) that are "weak". However, choosing the right threshold is a problem in itself. In the interest of simplicity, we calculate the median edge weight in each network and use that as a threshold. The median is preferable to an arbitrary threshold (such as 0.9 or 0.8 edge weight) and is also preferable to the mean as the latter is often skewed. After applying the threshold, we assume the remaining edges as representing meaningful relationships.

8.4.4 Overview of Analysis

We offer two types of analysis. One where we examine an individual network's properties and second where we compare two networks. In some cases we compare

[2]Most votes required Yay/Nay decisions. There were also a few Guilty/Not Guilty decisions.

corresponding profile and vote networks, in others we compare alternative profile networks.

Comparing two networks is essentially an instance of the more general Graph isomorphism problem in Computer Science. The problem consists of finding a bijective (one-to-one and onto) mapping from vertices in one graph to the other such that they are identical. Two graphs are isomorphic if such a mapping (known as an isomorphism) exists. Graph isomorphism is a hard problem and is known to be NP, although effective solutions exists for certain classes of problems (e.g., [74]). Another approach to compare two networks is to compute the edit distance. Graph edit distance is a measure of the cost of transformation of one graph to another via operations such as addition, deletion, substitution, etc, on nodes and edges. Each operation has some cost assigned to it. However, the computational complexity of edit distance for graphs is exponential in the number of vertices and is thus feasible only for small graphs (typically up to a dozen nodes) [100].

Since it is hard to design a general solution to the problem, we assess whether two networks are similar based on how similar they are with-respect-to certain characteristics. And since our networks are networks of people, we emphasize social networking characteristics in our comparison. We use the following properties to compare networks:

1. Edge weights in networks
2. Trends in strengths of ties
3. Trends in importance of nodes
4. Differences between Groups

We analyze the similarities and dissimilarities between profile-based networks and the vote networks using these properties. Information about their party status

is obtained from the Senate website. We analyze both general and issue-specific networks.

In the next section we provide the results organized by property. For each property we provide a more detailed description, followed by the results obtained and our analysis.

8.5 Results

8.5.1 Properties of Individual Networks

Table 8.2 shows the univariate statistics for mean Jaccard scores of General, abortion and immigration networks generated from voting data. We see that the mean strength of the relationships in the immigration network (0.400) is higher than the abortion network (0.256). Overall the mean score between senators is 0.346. The standard deviation is similar in all three networks. Note that the columns (in all tables) labeled General[3] refer to the pool of 6026 votes undifferentiated into categories.

	General	**Abortion**	**Immigration**
Mean	0.346	0.256	0.400
Std Dev	0.225	0.271	0.205
Minimum	0.014	0.000	0.015
Maximum	0.911	1.000	0.978

Table 8.2: Univariate Statistics for voting networks.

In comparison, table 8.3 shows the univariate statistics for mean similarity between senator profiles with stem and entity features for General, abortion and immigration networks. Note that unless otherwise mentioned, all profiles in this chapter

[3]We refer to a network created on the basis of all votes, regardless of issue, as a General network.

are generated using page-level data. Here we see that the mean strength of relationships between senators is higher in the abortion network (0.744 for stems and 0.724 for entities) than the mean for the immigration network (0.637 for stems and 0.467 for entities). Overall the mean similarity is 0.510 for the stem profiles network and 0.309 for the entity profiles network. It is clear from the table that similarity scores for profiles with entity features are significantly lower than for stem profiles. The standard deviation is similar for all networks except for the General network created from stem profiles.

	General		**Abortion**		**Immigration**	
	Stems	Entities	Stems	Entities	Stems	Entities
Mean	0.510	0.309	0.744	0.724	0.637	0.467
Std Dev	0.081	0.144	0.119	0.180	0.128	0.159
Minimum	0.238	0.061	0.273	0.052	0.203	0.081
Maximum	0.778	0.804	0.974	0.965	0.994	0.989

Table 8.3: Univariate Statistics for web (profile) networks

We obtain similar overall patterns of statistics for General networks for profiles using paragraph-level data. We also find that the average similarities are much lower for stem and entity profiles compared to scores for page-level profiles.

8.5.2 Comparing Edge Weights Across Networks

We now compare profile and vote networks based on their edge weights. We assume that an edge weight is a random variable (in the statistical sense) drawn from some underlying probability distribution. To simplify our analysis, we make another assumption that relationships among senators are independent of each other. In other words we assume that a relationship between senators A and B is not influenced by

A's relationship with another senator C. Consequently, the edge weights in a network can be considered as a sample of i.i.d random variables. Comparing a profile network and a vote network can then be considered as comparing two related samples drawn from two different or possibly the same probability distributions.

A standard statistical test to compare two related samples is the paired t-test. However, normality is a requirement for this test, which we cannot assume here. Therefore, we use a non-parametric version of this test, viz., the Wilcoxon Signed Rank test. This test involves comparing the differences in the edge weights. The null hypothesis is that there is no difference between the samples drawn from the two distributions and any perceived difference is due to chance alone. The test statistic W is calculated as:

$$W = \sum_{i=1}^{n} \phi_i R_i \tag{8.3}$$

where n is the number of scores used, ϕ_i is an indicator function for the sign of the difference (+ or -), and R_i is the rank according to the difference. Tied scores are assigned a mean rank and scores where the difference is 0 are ignored. As the number of scores (n) increases the distribution of all possible values of W approximates a normal distribution. For $n > 10$ the approximation is close enough to calculate the z score, which is calculated as:

$$Z = \frac{(W \ - \ \mu_W) \ \pm \ 0.5}{\sigma_W} \tag{8.4}$$

where μ_W is the mean and σ_W is the standard deviation. The ± 0.5 is a correction for continuity and is '-0.5' when W is greater than μ_W and '+0.5' otherwise. The significance of the Z value can be determined by a looking up the critical values in the z table. Table 8.4 shows the results of Wilcoxon test for comparing edge weights in voting and profile networks.

	General		**Abortion**		**Immigration**	
	Stems	Entities	Stems	Entities	Stems	Entities
W	8277629	-1491089	11743731	11413045	10208645	4007532
N	4949	4950	4949	4950	4950	4950
σ_W	201039.49	201100.42	201039.49	201100.42	201100.42	201100.42
Z	41.17	-7.41	58.42	56.75	50.76	19.93

Table 8.4: Results for Wilcoxon signed rank test.

Here, W is the test statistic, N is the number of scores (differences), σ_W is the standard deviation of W, and Z is the z score. As per the z table, for a 2-sided test, at the 0.05 significance level, we would reject the null hypothesis if $Z \geq 1.96$. We see that, ignoring the sign, all the z scores are greater than this critical value. Thus, we can conclude with very high certainty that the difference in the edge weights between the vote networks and corresponding networks is not due to chance and consequently the distributions of the edge weights in the vote networks and the corresponding profiles networks are not the same. We also apply this test for comparing networks created from profiles that use paragraph-level data. Consistent results are obtained.

	General		**Abortion**		**Immigration**	
	Stems	Entities	Stems	Entities	Stems	Entities
W	697861	-1237397	3695050	3220212	1657876	481636
N	3596	3608	3931	3516	3590	3585
σ_W	124525.84	125149.59	142323.56	120394.13	124214.35	123954.97
Z	5.60	-9.89	25.96	26.75	13.35	3.89

Table 8.5: Results for Wilcoxon signed rank test for filtered networks.

Table 8.5 shows the results of the Wilcoxon test for filtered profile and vote

networks. We remind the reader that as explained in section 8.4.3 in filtered networks edges retained have a weight greater than the median weight of edges in the full network. Again we see that all z scores are greater (ignoring the sign) than the critical value. Therefore, we conclude that the differences between edge weights in filtered profile and vote networks are also not due to chance and that although the two networks have the same nodes, they differ significantly in their edge weights.

8.5.3 Comparing Trends in Strengths of Ties

From the previous set of results, we have established that profile networks and vote networks are different in their edge weights . This is actually not surprising given that the Web and the Senate are two very different forums. However, they may still display similar trends in certain aspects. We are specifically interested in analyzing trends in senator relationships. Recall that one of our goals is to ascertain whether senators with similar voting patterns also have similar web profiles and vice-versa. Determining the correlation between the edge weights in the profile network and edge weights in the vote network would help us answer this question. A high correlation score would mean that the edge weights in the profile network follow the same trend as in the vote network. This may lead to a better understanding of the vote network via the profile network.

We first determine whether the type of association between the edge weights from the two networks is linear or non-linear by visually plotting the data. Figures 8.1 and 8.2 show the plots between edge weights from the General vote network and General stem profile and entity profile networks respectively. Clearly we see a non-linear relationship, irrespective of the type of features in the profiles. We also create plots (not shown here) for abortion and immigration networks and find similar non-linear associations.

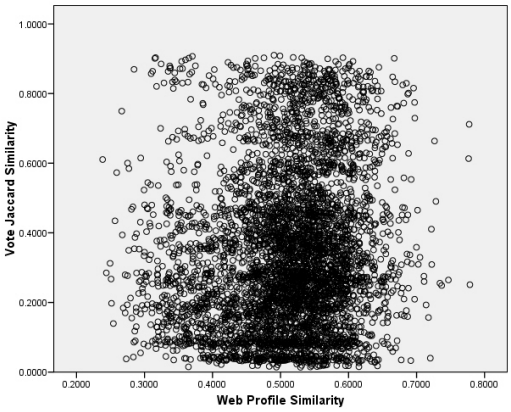


Figure 8.1: Profile Similarity vs. Vote Similarity. Profiles contain stem features.

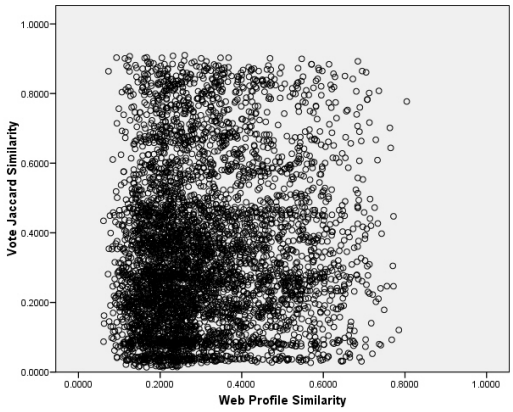


Figure 8.2: Profile Similarity vs. Vote Similarity. Profiles contain entity features.

Thus we use a non-linear measure of correlation. Specifically, we compute the correlation ratio (η), which is the ratio of the differences between networks to the overall (between and within) differences. Note that if an association is linear, the correlation ratio is the same as the Pearson correlation coefficient. Additionally we compute η^2, which denotes the percent of variance in the dependent variable explained by the independent variable. Since η is asymmetric, we compute its value in both directions, i.e., assuming that profile network weights constitute the independent variable and vote network weights are the dependent variable and vice-versa. Table 8.6 shows the values of η and η^2 in both directions.

		General		**Abortion**		**Immigration**	
	Direction	Stems	Entities	Stems	Entities	Stems	Entities
η	P $\rightarrow$ V	0.696	0.792	0.756	0.772	0.795	0.832
η^2	P $\rightarrow$ V	0.484	0.627	0.572	0.596	0.631	0.692
η	V $\rightarrow$ P	0.847	0.852	0.466	0.478	0.500	0.535
η^2	V $\rightarrow$ P	0.718	0.726	0.217	0.228	0.250	0.287

Table 8.6: η and η^2 values between edge weights in vote (V) networks and profile (P) networks.

First, considering the results where profile network weights are the independent variable, we see that the correlation ratios are fairly high throughout, especially for the immigration networks. More importantly, the η^2 values indicate that in the General network as well as for issue-specific networks, over half the variance in the vote network can be explained by the corresponding profile network. We also see that the correlation values are higher for entity profiles than stem profiles. Interpreting correlation scores is always tricky and completely depends upon the context. A score of 0.7 in one context may be considered high while the same score may be low in another context. Typically, it is much harder to obtain high correlation scores for social data, where the variability in the data is usually larger, in contrast to say scientific experimental data. Thus, we believe that the η and η^2 scores we obtain above indicate significant associations between vote networks and corresponding profile networks. While correlation ratios do not imply causality, to directly answer the question we asked above, we can say that there is some evidence that a high similarity in web profiles may mean a significant overlap in voting patterns.

Now, considering results in the opposite direction, i.e., when vote network weights are the independent variable, the correlation ratio and η^2 value for General networks is high (higher than in table 8.6) but is low for issue-specific networks. This means that a large proportion of the variance in weights in the General profile network can be explained by the corresponding edge weights in the vote network. However, this is not true for the issue-specific networks. This may be due to the considerably small number votes on specific issues (82 votes on abortion and 139 on immigration) compared to the total number of votes (6026). Another reason could be that the weights in issue-specific profile networks are too variable which makes it harder to explain them. Further research is required in this regard to gain a better understanding of the differences between issue-specific networks and the General

network in this context. Again to directly answer the question we asked above, the correlation scores we obtain provide some evidence that a significant overlap in voting patterns may mean a high similarity in web profiles in the general case but not for specific issues. Since we are interested in trends that span all pairs of senators, we do not do this analysis for filtered networks.

Overall, our results indicate that edge weights in profile networks may be used to reliably predict corresponding edge weights in vote networks. This provides motivation for exploring non-linear regression prediction models in future research.

8.5.4 Comparing Trends in Importance of Nodes

Often times one is interested in knowing *who is the most important person in a social network?* or in other words *who is the most connected?* More generally we may ask, relatively speaking, are senators equally important across the voting and profile networks?

There are various measures of *centrality* in SNA designed for answering these kinds of questions, such as Degree Centrality, Betweenness Centrality, etc. Degree Centrality is the simplest and measures how well connected a node is in a network. More formally, in a binary network it is defined as the number of edges incident upon a node. In a weighted network it is defined as the sum of the weights of the edges incident upon a node. In a social network the most well-connected people arguably have the most influence or importance.

We compare our profile networks with the vote networks on the basis of the ranking of senators by degree centrality. We compute the Spearman's correlation coefficient (ρ) to determine the association between the ranks in the two networks. A high rank correlation score would mean that the importance of senators is preserved across the two networks. Table 8.7 shows the correlation between the ranks of senators

across unfiltered and filtered profile and vote networks[4].

Network	**General**		**Abortion**		**Immigration**	
	Stems	Entities	Stems	Entities	Stems	Entities
Unfiltered	0.174	0.135	0.238*	0.169	0.258**	0.255*
Filtered	0.214*	0.123	0.205*	0.151	0.242*	0.257**

Table 8.7: Degree centrality rank correlations across unfiltered and filtered networks.

From the table, for unfiltered networks, we see that correlation values are low but positive for General as well as issue-specific networks. This implies a weak association between the importance of senators in vote networks and corresponding profile networks. The highest rank correlation is obtained for immigration profile networks (both stems and entities). Both correlation values are significant at the 0.01 level. The correlation values are low and not significant for the General networks. In the case of abortion, the correlation value for profiles with stem features is significant. Generally, speaking, the difference in features, i.e., stems vs. entities, does not have a major effect. For filtered networks, we see consistent results, i.e., correlation values are low but positive. Only correlation scores for stem profiles are significant except for the immigration network where the score for entity profiles is also significant. Again, the highest correlation scores are for immigration networks. We also calculate scores for profile networks in which profiles are created from paragraph-level data and find similar weak associations.

Thus we conclude that the influence of a senator with respect to other senators only mildly carries over from the vote network to the profile network. A senator may be more important or highly connected to others based on the votes cast in the Senate

[4]Here and in subsequent tables * denotes significant at 0.05 level and ** denotes significant at 0.01 level.

but that may not be true based on the information available on the Web.

8.5.5 Comparing Differences in Groups

In society people generally affiliate themselves with different groups and these affiliations determine their actions to a certain extent. Affiliations can be on the basis of geographic location, language, gender, etc. Additionally, in the case of senators, Party affiliation plays an important role. 99 of the 100 senators in the Senate belong to either the Democratic Party or the Republican Party. Many times a Party takes strong measures to ensure that members cast their vote in a certain way.

We analyze the differences between groups of senators belonging to the two parties in voting and profile networks. Specifically, we determine the difference between the average strength of the relationships between senators in each group. In other words we look at how cohesive the two parties are. We also determine if the difference between the two parties in the vote network is mirrored our profile networks.

	General		**Abortion**		**Immigration**	
	Dem	Rep	Dem	Rep	Dem	Rep
Mean	0.581	0.234	0.546	0.107	0.582	0.264
Std Dev	0.238	0.129	0.326	0.141	0.193	0.124
Minimum	0.038	0.035	0.014	0.000	0.151	0.078
Maximum	1.000	0.513	1.000	0.704	1.000	0.580
Difference	**0.347****		**0.438****		**0.318****	

Table 8.8: Mean similarities for Democrat and Republican senators from voting data and differences between the two groups.

Table 8.8 shows the mean strength (Jaccard score) of the relationships between Democrats and Republicans in the vote network. There are 45 Democratic senators

and 55 Republican senators. We see that in the General network and the issue-specific networks, the average similarity for Democrats is significantly[5] higher than Republicans. This suggests that the Democrats vote along the same lines far more frequently than the Republicans in general and on specific issues and are thus a more cohesive group.

Table 8.9 shows the mean similarity scores for Democrats and Republicans in the profile networks. Here Democrats form a significantly more cohesive group in the General and immigration networks. Unlike, with the voting data, we find that the cohesion between the two groups is not significantly different on the issue of abortion. The results are consistent across stem and entity features in profiles, although the differences for the latter profiles are higher. Overall, the highest mean similarity can be seen in the abortion network. This means that on this particular issue, the information present in web pages retrieved for senators is very consistent, in both the text and named entities, across all senators within each party. We also analyzed General networks created from profiles using paragraph-level data and found statistically significant differences between the two parties, for both stem and entity profiles.

Now we present results for filtered networks. Table 8.10 shows results for filtered vote networks. As with the unfiltered networks, we see that the mean similarity for Democrats is significantly higher in all three networks. We also see an increase in the difference between the two groups in the General and immigration networks over the difference in the corresponding unfiltered networks. The difference between the two groups in the abortion network remains roughly the same.

Table 8.11 shows the mean similarities and differences for Democrats and Republicans in filtered profile networks. As with the unfiltered networks, we see a

[5]Throughout this experiment, we calculate significance values using a bootstrap procedure which involves iterating over random sub-networks generated from the original networks. We compute the test statistic for these sub-networks. We then compute the probability of getting a score as small as we observe for the original networks. This constitutes the significance value. We use UCINET [24], a social networking package, for this analysis.

	General				**Abortion**				**Immigration**			
	Stems		Entities		Stems		Entities		Stems		Entities	
	Dem	Rep	Dem	Rep	Dem	Rep	Dem	Rep	Dem	Rep	Dem	Rep
Mean	0.525	0.498	0.256	0.207	0.841	0.802	0.810	0.773	0.687	0.619	0.511	0.412
Std Dev	0.045	0.055	0.140	0.047	0.081	0.111	0.106	0.147	0.081	0.098	0.108	0.097
Minimum	0.347	0.331	0.099	0.111	0.567	0.480	0.356	0.167	0.519	0.334	0.312	0.216
Maximum	0.628	0.568	1.000	0.302	1.000	0.934	1.000	0.909	1.000	0.756	1.000	0.574
Difference	**0.027****		**0.048****		**0.039**		**0.037**		**0.068****		**0.100****	

Table 8.9: Mean similarities for Democrats and Republicans from web (profile) data and differences between the two groups.

	General		**Abortion**		**Immigration**	
	Dem	Rep	Dem	Rep	Dem	Rep
Mean	0.546	0.142	0.528	0.058	0.543	0.108
Std Dev	0.297	0.182	0.350	0.151	0.263	0.187
Minimum	0.000	0.000	0.000	0.000	0.000	0.000
Maximum	1.000	0.0.513	1.000	0.704	1.000	0.580
Difference	**0.404****		**0.470****		**0.435****	

Table 8.10: Mean similarities for Democrats and Republicans from filtered voting data and differences between the two groups.

significant difference between the two groups in the General and immigration networks. Also, as above the differences for the abortion networks are not significant. The type of features in the profile also has an impact in this analysis. The difference between the two groups is more pronounced for stem profiles compared to entity profiles. In the case of the General networks, there is a significant difference between the two groups when similarities are based on stem features as opposed to named entity features. This also highlights one of the strengths of our approach as profiles offer different options for analyzing relationships between entities or groups.

To conclude, our results show that the differences between the Democrats and Republicans in the General and immigration vote networks are mirrored in our profile networks. This is not true for abortion. More generally, differences between groups as evidenced in a "real-world" network are preserved in networks of profiles created from web data. Thus, we can use profile networks to gain an understanding of what binds entities (senators) together in a group. For example, we can analyze the top shared features in the profiles of Democrats and Republicans to identify what they have most in common. This information may help in gaining a better understanding

	General				**Abortion**				**Immigration**			
	Stems		Entities		Stems		Entities		Stems		Entities	
	Dem	Rep	Dem	Rep	Dem	Rep	Dem	Rep	Dem	Rep	Dem	Rep
Mean	0.330	0.251	0.220	0.204	0.456	0.385	0.440	0.408	0.444	0.310	0.344	0.0.263
Std Dev	0.144	0.160	0.143	0.131	0.212	0.250	0.260	0.237	0.185	0.228	0.168	0.186
Minimum	0.000	0.000	0.003	0.003	0.000	0.000	0.000	0.000	0.000	0.0.000	0.000	0.000
Maximum	0.596	0.489	0.458	0.408	0.696	0.704	0.689	0.724	0.679	0.629	0.597	0.549
No. of Obs	45	55	45	55	45	55	45	55	45	55	45	55
Difference	**0.079***		**0.016**		**0.072**		**0.032**		**0.134****		**0.080***	

Table 8.11: Mean similarities for Democrats and Republicans from filtered web (profile) data and differences between the two groups.

of the ties between senators within these two groups based on the votes they cast.

8.6 Discussion

In this chapter we have presented an exploratory application of our profile-based approach to analyze networks of US senators. We compared networks created from web data with networks created from voting data obtained from the US Senate. We considered General networks as well as networks specific to two issues, viz., abortion and immigration.

Our results show that vote networks and web profile networks are different from each other in terms of edge weights. This is not surprising since the underlying data sources, i.e., Senate votes and the Web, are very different in many aspects. However, our results also show that it may be possible to make predictions regarding higher-level trends in vote networks using profile networks. For example, we find a strong non-linear relationship between the strength of relationships in the profile networks and vote networks. Our analysis of the differences between the two major parties in the Senate yields interesting results. We find that the Democrats as a group are significantly more cohesive than Republicans in terms of their voting patterns in the Senate. Our profile networks demonstrate this property in general and also for the issue of immigration but not for abortion.

In future research, we plan to explore other types of networks. Specifically, we will analyze and compare profile networks for countries with networks created from UN General Assembly votes. In this context, we can also analyze different groups, such as developing and developed countries, etc. We are also interested in analyzing changes in people profiles over time. In the senators context, we can also analyze changes in web profiles due to major events such as Congressional elections.

CHAPTER 9
APPLICATION: PROFILES FOR EXPERT SEARCH

In June 2007 we were presented a ready opportunity to apply our profile-based approach to another problem, viz., expert search. Specifically, this was the Expert Search task of the 2007 Text REtrieval Conference (TREC) Enterprise Track. The goal of the task was to identify experts for given topics using data from a collection of documents. In response, under the direction of Prof. Srinivasan, a student research group was formed to tackle the task. Our group developed a pipeline process that essentially consisted of processing the documents and creating topic and expert representations that were used to rank experts for each topic. Figure 9.1 illustrates this process[1].

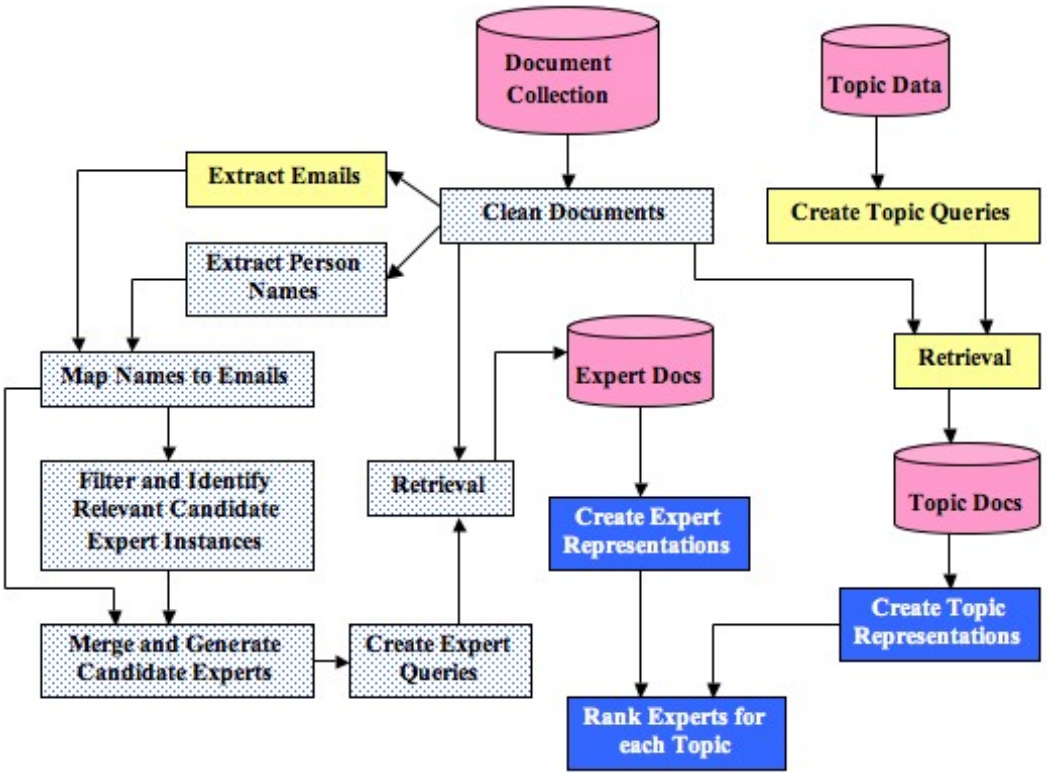


Figure 9.1: UIowa pipeline process for 2007 TREC Expert Search task.

[1]In the figure, boxes with yellow background denote steps implemented by other group members. All other steps (solid and dotted blue background) were implemented by us.

We used two different representation methods, one based on topic models (probabilistic representations) and the other based on our topic profiles. The TREC guidelines allowed us to submit up to four different runs. Three of the runs were allocated to probabilistic model based approaches[2] and one to our topic profile-based approach. All the runs were due to be submitted by August 9, 2007.

Below we describe the individual steps in the pipeline process and present results obtained for our profile-based approach applied to this "live" task. This is a joint effort and solutions for some of the steps in the pipeline were developed by other members in our group. We provide references at appropriate places in the text below. Overall in this chapter we present only our profile-based approach.

9.1 Background

TREC is an annual conference organized by the National Institute of Standards and Technologies (NIST). TREC is organized as a series of Tracks, each consisting of one or more specialized problems in Information Retrieval. Example tracks include the Enterprise track and Genomics track. Various research groups at Universities and private companies throughout the world enroll to participate in one or more tracks. The Track organizers define the tasks and make available the necessary data. The Enterprise track was introduced in 2005. One task of this track is expert search. Given a diverse set of topics and optionally a list of candidate experts, the goal is to find and rank the experts for each topic based on the information within a collection of documents. This document set is also provided by the track organizers. The overview documents for the 2005 [52] and 2006 [117] enterprise tracks provide in-depth descriptions of the task and the data in the respective year. An outline of the 2007 task can be found on the track website[3].

[2]Due to some problems only one of these runs was submitted.
[3]http://www.ins.cwi.nl/projects/trec-ent/

9.2 Related Research

Expert search is a well known problem in Information Retrieval and various academic and commercial solutions have been developed. Simple attempts in this area mainly consist of a search interface to information that people provide about themselves, such as keywords describing expertise. E.g., AllExperts[4] is a popular online forum where people can search for volunteer experts to answer questions in different areas. Volunteer experts themselves provide information regarding their expertise. More sophisticated approaches have been proposed to automatically identify experts modulo a document collection. E.g., P@NOPTIC Expert [53] is a commercial expert finding system to find a list of experts in a certain area based on information in a document collection.

Since 2005, the TREC expert search task has provided a fertile ground for research in this area. In 2005, 9 groups participated in the task and in 2006, 23 groups participated. The top ranked group in 2005 [63] created pseudo documents for candidate experts using a variety of features extracted from relevant documents, including the text, inlink anchor text, metadata and bigrams. These pseudo documents were then indexed and ranked for each topic query in the style of traditional document retrieval. They also utilized the html tags present in pages to improve performance. In 2006, for the same set of documents but for different topics, there was a marked improvement in performance in general. The top ranked submission in this year [146] used a 2-stage language model: a relevance model for retrieving documents and a co-occurrence model for searching experts. They utilized a PageRank like measure in their relevance model to determine a document's authority. They also used window-based co-occurrence between names and query terms to identify experts. They created specific templates to take into account structural differences between different types of documents, such as technical reports, emails, etc. These templates were used to

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

```
<top>
<num>CE-001</num>
<query>genetic modification</query>
<narr>
Over arching information on gene technology /
biotechnology. Specific pages on certain GM (e.g.
cotton).
</narr>
<page>CSIRO135-03599247</page>
<page>CSIRO141-08973435</page>
<page>CSIRO141-07897607</page>
</top>
```

Figure 9.2: Example topic for TREC 2007 Expert Search task.

determine the weight of a name in a document based on its location. E.g., the name of an author of a technical report is assigned high weight.

9.3 Data Description

9.3.1 Document Collection

The document collection consisted of pages obtained from the websites of different departments within the Australian Commonwealth Scientific and Research Organization (CSIRO). The CSIRO corpus consists of approximately 370K pages crawled from different website under the csiro.au domain. Both text and non-text documents, including pdf, doc, ppt, ps, and xls were crawled. The non-text documents were included in the corpus after being converted to html format.

9.3.2 Topic Set

A small training set of 10 topics was provided and as well as a separate set of 50 topics on which the final submissions would be judged. For each test topic, the query, a brief narrative and up to 3 relevant documents were given to participants. An example topic is shown in figure 9.2.

9.3.3 Candidate Experts

A key element of the 2005 and 2006 expert search tasks was a list of candidate experts given to participants. In contrast, in the 2007 task, no list of candidate experts were given. Participants first had to identify candidate experts for each topic and then rank them.

9.4 Data Pre-processing

9.4.1 Email and Name Extraction

We pre-process the corpus in the following way. Firstly we remove the header information from each document as well as all non-ASCII characters. Next we identify all email addresses in the documents using a general pattern represented by the following regular expression:

\b([a-zA-Z0-9._%+-]+@(?:[a-zA-Z0-9-]+\.)+[a-zA-Z]{2,4})\b[5]

We then extract all the named entities in the documents using a named entity tagger. We mentioned previously (in section 5.1.5) that we compared the ClearForest tagger with the Stanford Named Entity Recognition (NER) system and found the former to be more accurate and therefore preferable. However, the free version of the system can process only 100 KB at a time and all the information has to sent and received over the internet. Also, the system has a prohibitive six-figure cost. Thus, we use the freely available Stanford NER system to extract named entities. We limit the extraction to names of people entities.

9.4.2 Name to Email Mapping

We map named entities to email addresses by applying various pattern matching rules. Table 9.1 shows the different patterns we used, in the form of PERL regular

[5]Thanks to Bob Arens for providing the regular expression.

$pat0 = /^$ent@/;
$pat1 = /^$fname$lname@/;
$pat2 = /^$fname[.|-|_]$lname@/;
$pat3 = /^$lname$fname@/;
$pat4 = /^$lname[.|-|_]$fname@/;
$pat5 = /^$finitial$lname@/;
$pat6 = /^$finitial[.|-|_]$lname@/;
$pat7 = /^$lname[.|-|_]$finitial@/;
$pat8 = /^$fname$linitial@/;
$pat9 = /^$lname@/;
$pat10 = /^$fname$minitial$lname@/;
$pat11 = /^$fname[.|-|_]$minitial[.\|-\|_]$lname@/;
$pat12 = /^$finitial$minitial$lname@/;
$pat13 = /$ent@/;

Table 9.1: Regular expressions to map names to email addresses.

expressions, in the order we apply them[6]. To explain a few of them, consider the following examples:

- John Malone Smith => johnmalonesmith@∗ (first pattern)
- John Malone Smith => johnsmith@∗ (second pattern)
- John Malone Smith => john.smith@∗ (third pattern)
- John Malone Smith => john-smith@∗ (third pattern)
- John Malone Smith => john_smith@∗ (third pattern)

[6]Here $fname denotes first name, $lname denotes last name, $finitial denotes first initial, $linitial denotes last initial, $minitial denotes middle initial, and $ent denotes the full name.

Note that one name may map to multiple email addresses and vice versa. We limit mapping to entities and emails that occur in the same document to avoid ambiguity. The final output of this step is a corpus-wide list of name-email mappings and unmapped names & emails. In our notation we refer to each member of this list as a *person instance*. From the CSIRO collection we get 1.5 million person instances.

The first task is to identify *candidate experts* for each topic. The only clue given by the task organizers in this regard was that relevant experts have emails that end in '@csiro.au'. Based on this we filter our list of 1.5 million person instances and retain 33,803 instances and then use a rule-based merging algorithm[7] to merge all the different instances. The final output is a list of 3312 candidate experts, each of which has at least one email ending in '@csiro.au'.

9.5 Profile-based Strategy

Our general methodology is to first create profiles for each topic and for each expert. For a topic, we then rank experts based on the similarity of their profiles. We submit this ranking as our response to the TREC task. Thus our first task is to identify relevant documents for topics and experts.

9.5.1 Retrieving Documents for Topics

We first index the corpus[8] and then retrieve documents using word-based boolean queries to represent each topic[9]. We create initial topic queries from the given query and narrative. Then we expand[10] these queries using the most frequent metadata terms present in the given relevant documents. We consider only DC metadata terms, such as DC.Subject and DC.Keywords. If relevant documents do not have

[7] Refer to appendix B for algorithm pseudo code and description.

[8] Corpus indexed using Lucene [2]. Thanks to Bob Arens.

[9] Topics provided by the organizers are more general than entities and thus phrase-based queries are not suitable.

[10] Thanks to Ha Thuc Viet for providing expanded topic queries.

any metadata terms then we retrieve documents for initial topic queries and then expand queries using metadata terms present in top ranked retrieved documents. We use the expanded queries to retrieve documents.

9.5.2 Retrieving Documents for Experts

We define relevance somewhat loosely in that a document is relevant if it contains an instance of an expert. For example, if a document contains the name or email id of an expert then it is considered relevant for that expert. For each expert we identify all relevant documents in the corpus.

9.5.3 Build Topic and Expert Profiles

We build profiles for topics using the top 100 documents retrieved for expanded topic queries and expert profiles from 1000 random documents containing instances of the expert. Profiles are created from page-level data and contain stem features. We use the best weighting method ($wt2$) from previous experiments (refer to chapter 6).

Due to various constraints, most important of which was time[11], we were not able to create other types of profiles. For example, we did not extract entity features because of the various limitations imposed by ClearForest that would result in waiting a long time for thousands of documents to be tagged. We also did not create paragraph-level and segment-level profiles as they are too expensive in terms of time when a large number of documents need to be processed.

9.5.4 Ranking Experts

For each topic, we rank experts based on the similarity of their profiles with the topic profile. Since generating profiles for each candidate expert would be a time consuming process, instead we only consider candidate experts that are referred to

[11] We started working on this task in mid June and runs were due in early August.

in the top M documents. For 49 of the 50 topics we consider experts in M=25 documents. For the remaining topic we consider M=75 documents because there were no instances of candidate experts in the documents for M<75.

9.6 Evaluation Measures

The primary evaluation measures are Mean Average Precision (MAP) and Mean Reciprocal Rank at 1 (RR1). MAP is defined as the mean of the Average Precision (AP) for each topic, where AP is the average of the precision at each recall point (position of relevant expert) in a ranked list of experts. Formally:

$$MAP = \frac{1}{n} * \sum_{i=1}^{n} \frac{\sum_{j=1}^{m} P_j}{m} \tag{9.1}$$

where n is the number of topics, m is the number of recall points in a ranked list, and P_j is the precision at the j^{th} recall point.

Mean RR1 is defined as the average of the reciprocal of the rank of the first relevant expert retrieved for a topic. Formally:

$$RR1 = \frac{1}{n} * \sum_{i=1}^{n} \frac{1}{R_i} \tag{9.2}$$

where n is the total number of topics and R_i is the rank of the top-ranked expert for topic i.

Secondary evaluation measures are precision in the top 5 ranked experts (P@5), precision in the top 10 ranked experts (P@10), precision in the top 100 ranked experts (P@100), precision in the top 1000 ranked experts (P@1000), and precision in the top r documents (r-prec), where r is the number of relevant experts. Additionally b-pref [40] is used to evaluate how well a ranked list follows the preferred order (relevant experts before non-relevant experts). It is formally defined as:

$$b\text{-}pref = \frac{1}{R} \sum_{r} 1 - \frac{|n \text{ ranked higher than } r|}{R} \tag{9.3}$$

where R is the number of relevant experts, r is a relevant expert, and n is a member of the first R judged non-relevant experts retrieved.

9.7 Results

Table 9.2 shows the scores for different evaluation measures for our approach (labeled T1000NS). Our approach has a MAP of 0.2828 and RR1 of 0.4321. Based on additional feedback we received from the track organizers, our approach performs better than the median for 27 of the 50 topics in terms of Average Precision (AP) and has an AP of 1.0 for 4 of the topics. In terms of Reciprocal Rank our approach does better than the median for 19 of the 50 topics and has an RR1 of 1.0 for 14 topics. As per preliminary conference proceedings[12], our approach is ranked in the bottom half of the submitted runs, based on MAP. Among the submitted runs, the highest MAP score (0.4787) is obtained for a manual run.

	MAP	**r-prec**	**b-pref**	**P@5**	**P@10**	**P@100**	**P@1000**	**RR1**
T1000NS	0.2828	0.2480	0.6217	0.1640	0.1160	0.0168	0.0017	0.4321

Table 9.2: Results for 2007 Expert Search Task.

9.8 Discussion

In this chapter we presented the application of our profile-based approach for the task of identifying and ranking experts for certain topics of interest. We submitted a run to the 2007 edition of the TREC expert search task. Based on the limited information we have at this time about other submitted runs, our results appear encouraging, especially considering that we have applied our general approach to solve this problem without taking into account the specific characteristics of the corpus.

[12]only accessible to conference participants as of December 3, 2007

These have been shown to be very useful in the past. For example, a key aspect of the top ranked submission [146] in the 2006 edition of this task, was the use of specialized templates to take into account structural differences between different types of documents, such as technical reports and emails. We believe that by customizing profiles for the data at hand we can also get better performance.

It is important to keep in mind that the accuracy of our profiles in this experiment depends upon the accuracy of the preliminary steps in our pipeline. For example, document retrieval, name and email extraction, generation of mappings between names and emails, all play an important role in determining the final output. At this point we do not know about the error rate associated with each step of our pipeline. Thus a key area for future research will be to analyze the errors at each step in the pipeline and also analyze the effects of these errors on the accuracy of our profiles for this task. However, we are pleased that our profile-based approach was readily applicable to a new problem with a short preparation time.

CHAPTER 10
DISCUSSION AND FUTURE WORK

In this chapter we first provide a summary of this thesis, then provide a general discussion of what we have learned from this work, and finally outline avenues of future research.

10.1 Summary

In this thesis we have presented an approach for representing different kinds of topics using data from the Web. A topic profile is analogous to a synopsis of a topic and consists of different features that characterize it. Topic profiles are flexible in that they allow different combinations of features to be emphasized. They are extensible in that new types of features can be easily incorporated without having the make any modifications to the underlying logic. Topic profiles provide a natural framework to explore relationships between topics, as determined for example by profile similarity. A key point here is that relationships based on topic profiles do not depend upon explicit indicators such as co-occurrence of the topics in documents within a collection. Thus, implicit relationships can also be explored using profiles. As mentioned previously (in chapter 3) one of our long-term goals is the automatic discovery of novel hypotheses from web data. In this regard another key point is that the flexibility of profiles allows for analyzing different kinds of relationships between topics.

In chapter 2 we provided a broad overview of existing text and web mining research from the perspective of knowledge discovery. Compared to text mining in specialized domains, we find that web mining research is still at a preliminary stage. Thus one of our goals was to expand the scope of knowledge discovery research to the Web.

Our profile-based approach is designed to overcome important limitations in existing approaches. Most are designed only for topics representing entities, primarily people entities, and not all kinds of topics. In contrast, our profiles can be created for both entity topics and general topics as long as they can be represented by an appropriate search query. Existing approaches also make specific choices regarding some key factors such as the number of pages, level of data, term weighting method, and features extracted. We have systematically explored the relative value of these and additional options (chapter 3). Moreover, our profiles are general in that different choices along these dimensions can be accommodated. For example, Raghavan et al. [106] depend upon instance-level data to create entity models whereas in our approach we can use data from a variety of levels. Adamic and Adar [26] utilize certain features extracted only from the home page of people to generate representations for them. In our approach we can additionally consider multiple relevant pages.

A strong point in the way we have designed our research is that topic profiles provide an abstract framework that can be used to create different types of concrete representations for entities and topics. Different options regarding the number of pages or features extracted can be decided based on requirements of the problem as well as the characteristics of the data. Irrespective of the specific configuration, the underlying logic across these different types of profiles remains the same. A more detailed description of our profile framework and related properties can be found in chapter 4. This framework is a significant contribution of this thesis.

In this thesis we implemented several options for the key dimensions mentioned above. We offer standard approaches, such as page, paragraph and sentence-level data, $tf*idf$ weights, etc., and also contribute new approaches based on our intuition. For example, in an effort to distill relevant information from pages, we designed an algorithm to extract relevant segments from text. We also designed several new

term weighting methods, each based on a certain intuition. For example, the $wt2$ weighting method ($tf * [log(df) + 1]$) is based on the intuition that features that occur more frequently within documents and also across documents would be effective at characterizing a topic. Although it did not work as well as expected, we present an interesting and novel application of h-index for term weighting. H-index has been widely used for measuring people contributions and intuitively the problem of assessing the impact of people is similar to the problem of assessing the importance of terms. We also explore the use of the tag structure of html pages to modify the weights of terms, which has also not been considered in existing web mining approaches. Chapter 5 contains a detailed description of the general methodology for building profiles.

Our first experimental goal was to compared the different types of possible profiles. In chapter 6, we compared profiles generated from web data with profiles generated from Wikipedia. This research was informed by our preliminary work [114] published in the I3 workshop at WWW 2007. Our results show that page-level profiles with stem features assigned $wt2$ weights (described above) are best. Interestingly, $tf * idf$ does not perform well compared to our best strategy. Paragraph and segment level profiles perform equivalently to page profiles but it takes significantly longer to create these types of profiles. We also analyzed different sources of error and found that tools specifically designed for the Web perform well while tools designed for general text documents do not. This points to the need for specialized foundational tools for web mining, such as for sentence boundary detection, for web data.

In chapter 7, we compared different types of profiles based on their ability to predict known protein interactions mentioned in DIP, using data from the Web. We found that paragraph-level profiles with named entity features are the most accurate and are significantly better than most of the other types of profiles. This reinforces

our point that flexibility in representation is essential to deal with different kinds of problems. This is one of the key limitations in existing approaches and one that is overcome by our topic profile framework. A secondary goal of this thesis and particularly this experiment was to explore biomedical knowledge discovery using web data. Although our results are moderate these offer a reasonable starting point. Given the vastness of the Web, it is fast becoming a source of data and knowledge for domain specific applications.

Across both evaluations (Wikipedia and DIP) we found that using the tag information to modify term weights had a detrimental affect on performance. We are not discouraged by this result. The particular strategy that we used was adopted from research by Culter et al. [55] in which they use tags to improve document retrieval. We have already seen that the standard $tf * idf$ term weighting, which works well for document retrieval does not compare well with our new methods in the context of profiles. It is possible that we are seeing a similar pattern here. This motivates considering new strategies for future research, specifically designed keeping topic profiles in mind.

At this point, we remind the reader that our profiles are built using a pipeline process. Every step in the pipeline contributes towards the final profile. Improvement in any of the individual steps will likely lead to an improvement in the final profile. For example, increasing the accuracy in recognizing sentences in documents would lead to improved sentence-level profiles. In a similar vein, improvements in the named entity detection system would lead to better features in profiles. Using a pipeline process also allows us to add new modules at any point as was the case in our experiments with expert search (chapter 9).

One of the primary goals of knowledge discovery research is to facilitate analysis of implicit and explicit relationships between topics. In chapter 8 we presented

an application of our profile-based approach to analyze social networks, specifically networks of US senators. We compared networks from web data with "real-world" networks generated from voting data. We found that while profile networks and voting networks are different in certain aspects, they do share certain trends. For example, it is highly likely that senators who have a strong relationship in the profile network also have a strong relationship in the voting network. We also analyzed social groups of senators and found similar trends across profile and voting networks.

Finally, in chapter 9 we described an application of our approach to expert search. We came across an opportunity to participate in the 2007 TREC expert search task and in a short amount of time we were able to complete this work. This demonstrates the broad applicability of our approach, The goal of the task was to identify and rank experts for given topics. A final important point to underline is that TREC topics were much more general than "entities". This effort demonstrates that our framework is designed to handle both topics and entities. This is in contrast to many other web knowledge discovery research methods (e.g., [106, 31]) as they are limited to specific entities.

10.2 General Discussion

Our research in this thesis is motivated and informed by our research in biomedical knowledge discovery over the last few years. While there are significant challenges in biomedical text mining, such as ambiguity in biomedical terms, the availability of many high-quality information sources such as MEDLINE, DIP, and SwissProt, provides an ideal setting for text mining/knowledge discovery research.

In contrast, the Web is a much more difficult platform for knowledge discovery/web mining research. As a general resource it contains information on wide variety of subject areas. Almost anybody can publish their opinions online and these

are not necessarily reviewed for factual errors. In the biomedical domain a significant amount of information is curated and thus of very high quality. Also, useful secondary resources such as ontologies and entity databases (genes and proteins) are plentiful in the biomedical domain.

However, despite these problems the Web is among the the largest sources of information available today and offers a tremendous opportunity for new research. An advantage of the Web is that it encourages inter-disciplinary research. For example, it is unlikely that documents in PubMed would lead to an implicit connection between a biomedical entity and an entity from Physics. The Web contains information from both domains and thus it is more likely to provide such types of connections. The Web also encourages "social" research by offering a rich assortment of documents describing people, places, and other social entities. The availability of news articles and blogs available on the Web are an added attraction.

In this thesis we faced a number of significant challenges. Some of these stem from the nature of the problem of creating representations of topics from text documents and exploring relationships between topics. However, the most difficult challenges were primarily due to the nature of the pages on the Web, specifically the lack of structure. Web pages are semi-structured and to a certain extent rely on visual presentation to convey information. For example, a table in a web page groups related information together and conveys the meaning of the information through rows and columns. While the meaning may be apparent to humans, it is much more difficult for automatic algorithms to handle effectively.

Most text processing tools rely on rules based the presence of certain syntactic and linguistic cues to extract information from documents. However, in web pages many of the rules that apply to plain text documents do not hold. For example, a sentence boundary detection algorithm will mash together the text in each cell in a

table and output a large "sentence". Machine learning techniques have been used to train algorithms to recognize different structures in web pages that follow a specific structure (e.g,. from the same website) but this approach faces difficulties when applied to an arbitrary set of pages, which is what we consider in this thesis. Our approach consists of using intuition-based heuristics to impose structure at certain places in web pages. For example, we consider a row in a table to be a logically similar to a sentence and thus add a sentence delimiter at the end. Another related challenge included cleaning retrieved web pages by replacing non-ascii characters with colloquially used ascii characters (e.g., 'o' for 'ö'). This was necessary as some of the text processing tools we used, e.g., Hearst's TextTiling implementation, are not designed to handle UTF8 characters.

The design of experiments was another significant challenge that we faced in this work. We offer several options for different parameters such as level of data used, term weighting method, and features selected. It was not possible to compare all the different combinations of these options as there would have been too many. Thus, we created a sequential design wherein at each step we selected a particular parameter, compared profiles created from different options for this parameter, and selected the best option. All the other parameters were kept constant. The best option was then used in the next step where comparison would be on the basis of a different parameter. This design made this problem tractable.

10.3 Future Work

At the end of chapters 6, 7, 8 and 9 we provided specific future research directions for the problems addressed in those chapters. A major emphasis of future research will be on augmenting profiles with other types of features that we have not considered in this thesis. For example, for biomedical topics we can use the UMLS

ontology to help identify biomedical concept features from web pages. We believe that domain specific features will lead to a significant improvement in the performance of our profiles for problems such as predicting protein interactions. Additionally we plan to extract more structured features such as Relations. Relations provide an explicit acknowledgment that two concepts are connected. For example, the relation 'supported(Bush, Gulf War)' extracted from a sentence in some document denotes that Bush supported the Gulf War. Such features would allow us to build profiles that contain more structured information, which would improve the readability of profiles and would also probably lead to better precision than just word or stem features when analyzing relationships between topics.

Another area that we are interested in exploring in the future is the integration of ontologies with our profiles. An ontology is a data model that defines concepts and relationships between concepts. For example, the UMLS ontology contains biomedical concepts and defines relationships between these concepts. We believe that such pre-existing high-quality information can be used to improve the quality of information in profiles as well as detect relationships between topics that are not directly apparent from the overlap of concepts in their profiles text but can be inferred from the relationships defined in the ontology between profile concepts.

Finally, we are interested in implementing the Open and Closed discovery processes to hypothesize potentially new connections between topics based on information available on the Web. The Open and Closed Discovery processes developed by Swanson [125] are pioneering efforts in biomedical text mining and perhaps text mining in general. Preliminary research by Gordon et al. [67] demonstrates the feasibility of applying these methods to web data. However, these methods have not been fully explored in the web context.

To close, this research contributes to our long-term goal which is to be able

to represent arbitrary topics on the Web with topic profiles consisting of weighted features of different types. We see value in pursuing a higher level topical Web where the object (node) of interest is the topic and a link represents an inter-topic relationship. Such a Web has the potential to more effectively support individual information needs as well as the requirements of web mining applications seeking to discover novel connections between topics.

APPENDIX A
WEBKD INTERFACE AND SCHEMA

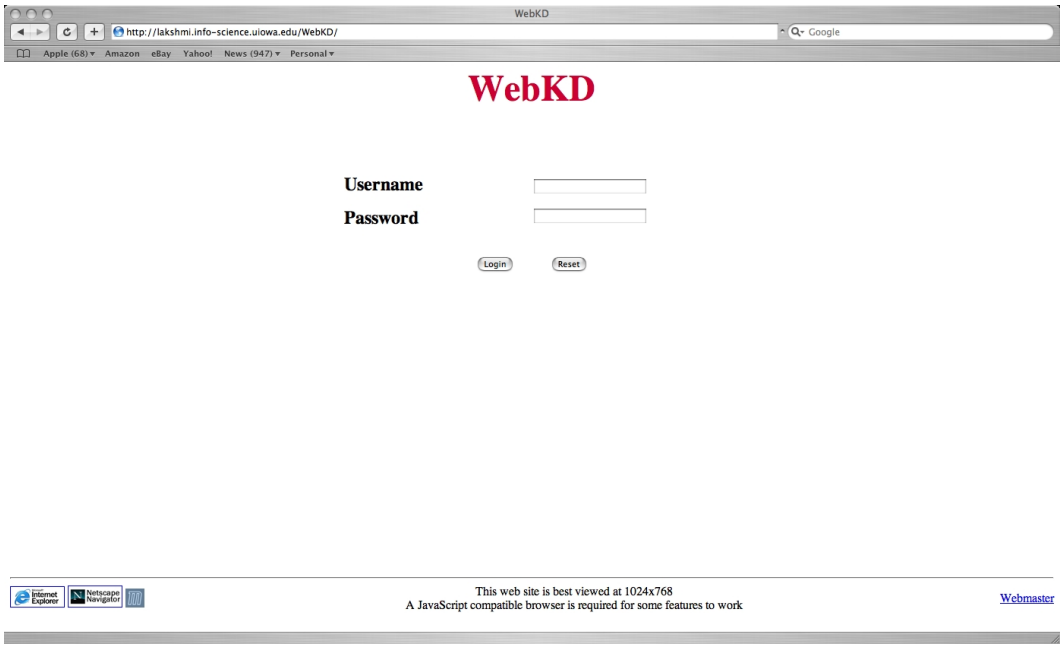


Figure A.1: WebKD Main Page

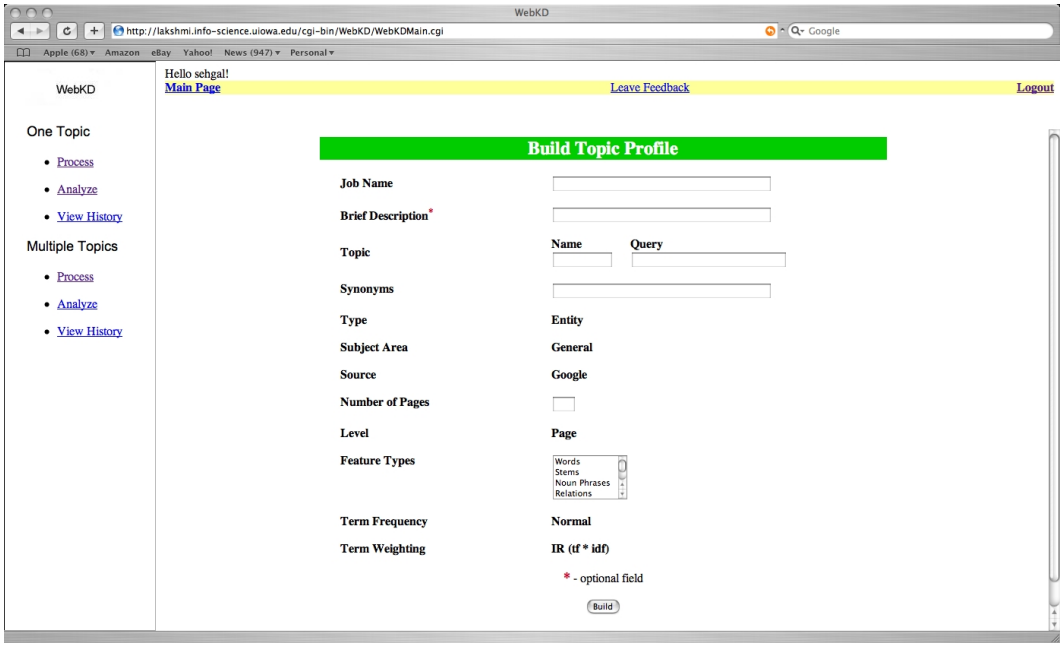


Figure A.2: WebKD One Topic Process - Basic Form

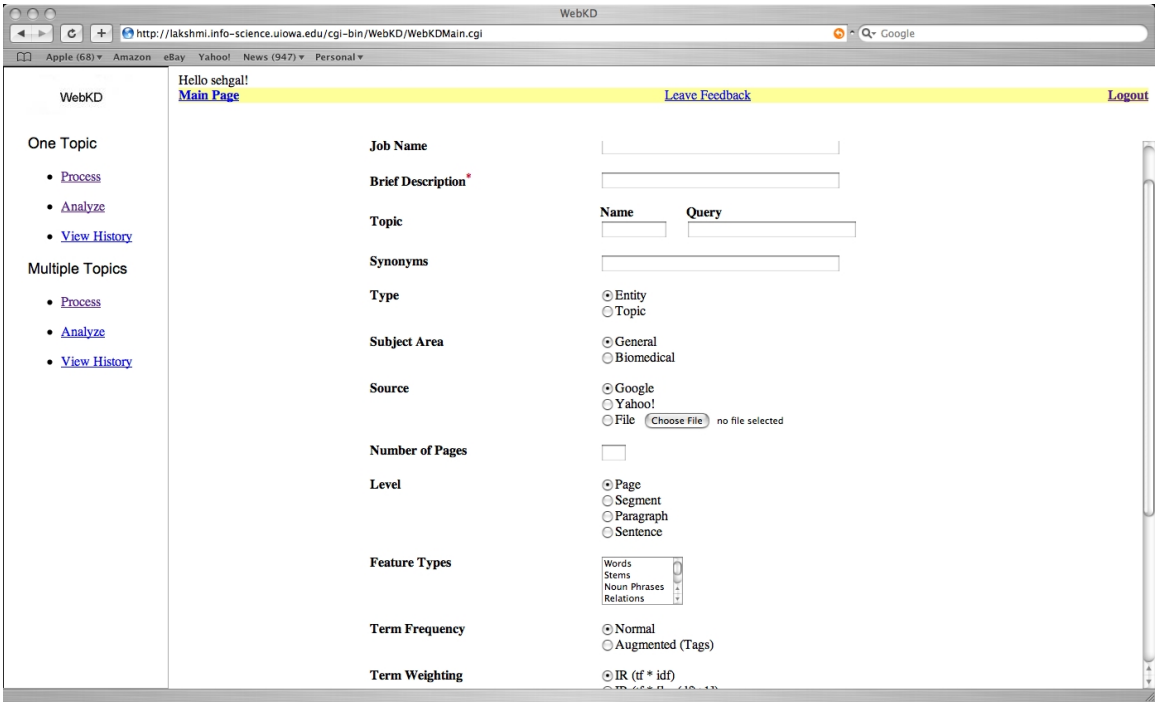


Figure A.3: WebKD One Topic Process - Advanced Form

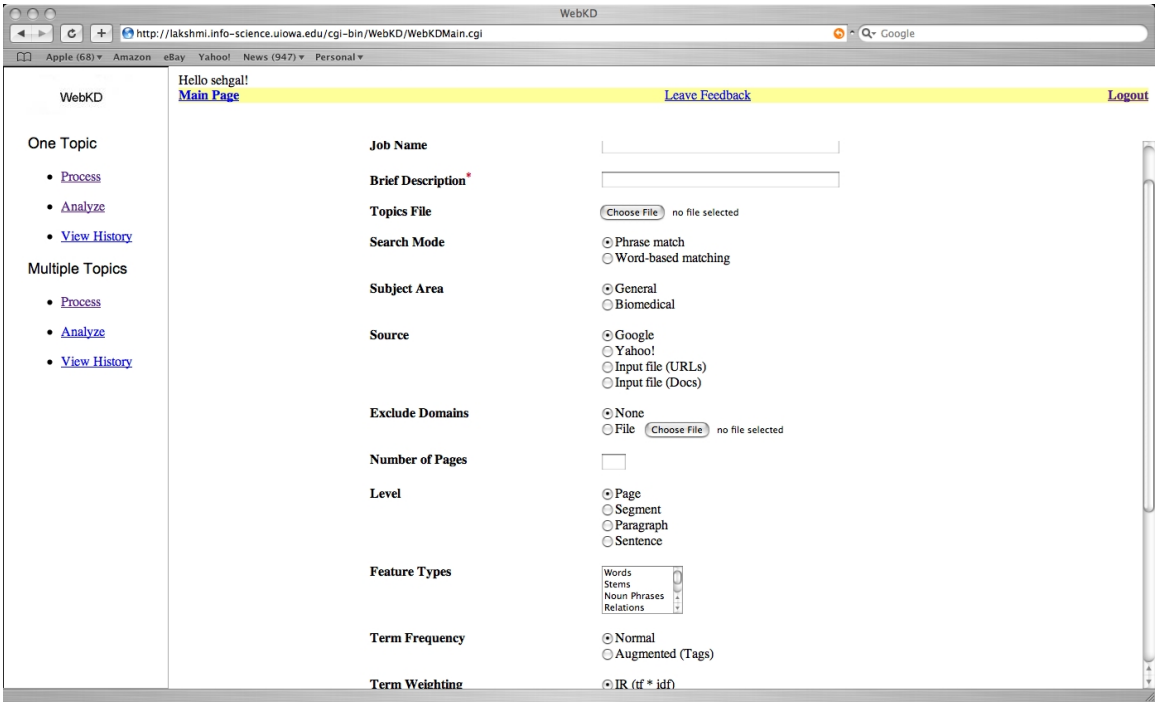
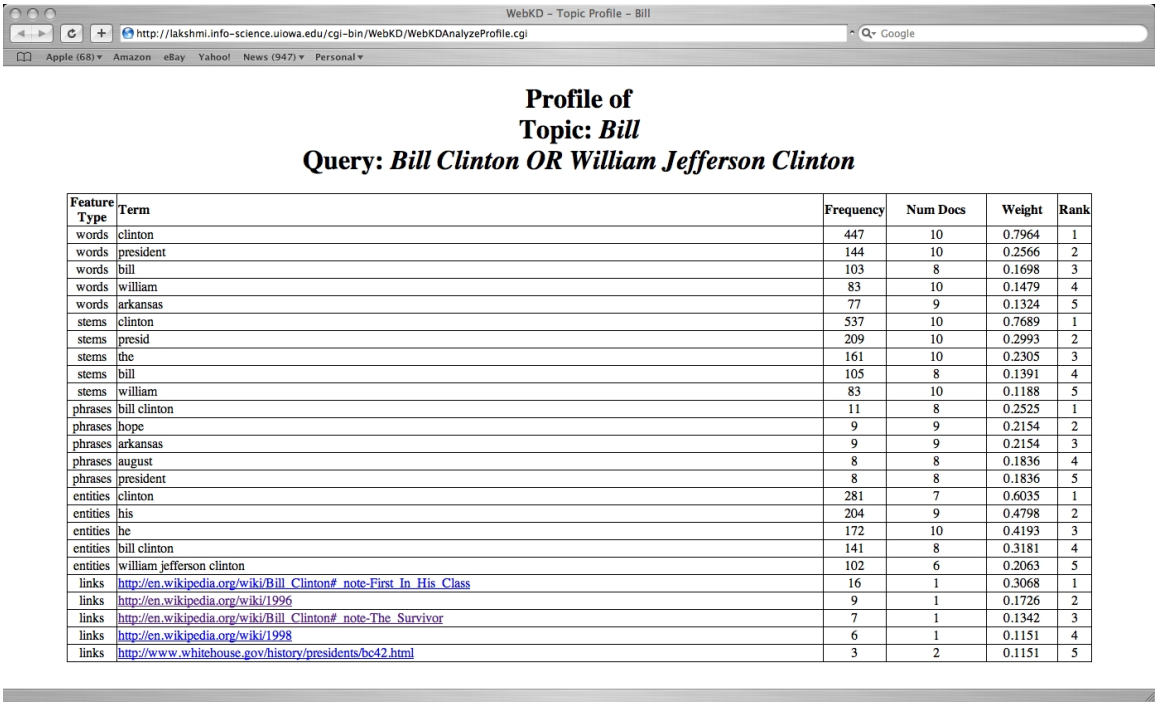


Figure A.4: WebKD Multiple Topics Process - Advanced Form



Profile of
Topic: *Bill*
Query: *Bill Clinton OR William Jefferson Clinton*

Feature Type	Term	Frequency	Num Docs	Weight	Rank
words	clinton	447	10	0.7964	1
words	president	144	10	0.2566	2
words	bill	103	8	0.1698	3
words	william	83	10	0.1479	4
words	arkansas	77	9	0.1324	5
stems	clinton	537	10	0.7689	1
stems	presid	209	10	0.2993	2
stems	the	161	10	0.2305	3
stems	bill	105	8	0.1391	4
stems	william	83	10	0.1188	5
phrases	bill clinton	11	8	0.2525	1
phrases	hope	9	9	0.2154	2
phrases	arkansas	9	9	0.2154	3
phrases	august	8	8	0.1836	4
phrases	president	8	8	0.1836	5
entities	clinton	281	7	0.6035	1
entities	his	204	9	0.4798	2
entities	he	172	10	0.4193	3
entities	bill clinton	141	8	0.3181	4
entities	william jefferson clinton	102	6	0.2063	5
links	http://en.wikipedia.org/wiki/Bill_Clinton#_note-First_In_His_Class	16	1	0.3068	1
links	http://en.wikipedia.org/wiki/1996	9	1	0.1726	2
links	http://en.wikipedia.org/wiki/Bill_Clinton#_note-The_Survivor	7	1	0.1342	3
links	http://en.wikipedia.org/wiki/1998	6	1	0.1151	4
links	http://www.whitehouse.gov/history/presidents/bc42.html	3	2	0.1151	5

Figure A.5: WebKD Topic Profile for *Bill Clinton*

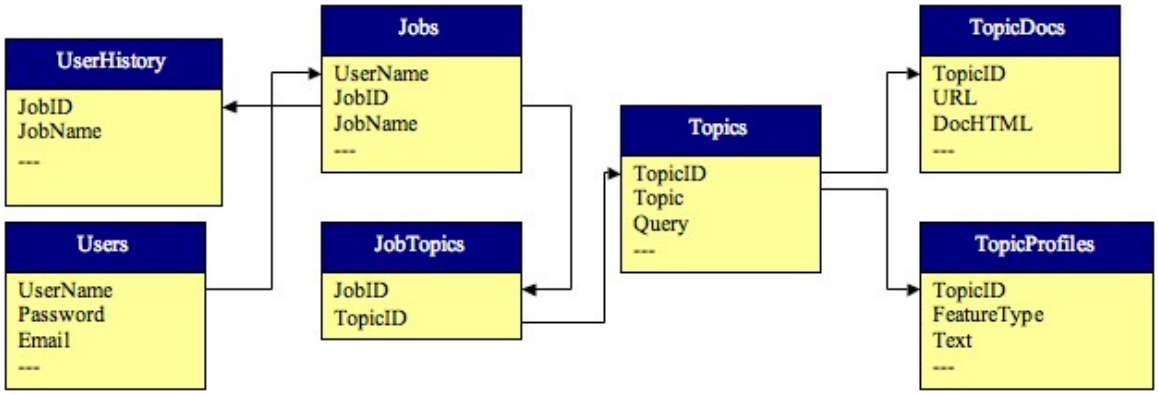


Figure A.6: WebKD Database Schema

APPENDIX B
EXPERT SEARCH MERGING ALGORITHM

```
1: Input: expert file
2: Input: entities file
3: Output: normalized experts file
4: var hash trecexperts // email is key
5:
6: for all expert ϵ expertfile do
7:    if expert.email ∉ trecexperts then
8:       add expert
9:    else
10:      merge expert
11:   end if
12: end for
13:
14: for all entity ϵ entitiesfile do
15:   clean entity
16:   done ← 0
17:
18:   if email ≠ 'Unknown' then
19:      if entity.email ϵ trecexperts then
20:         merge expert and add doc id
21:         done ← 1
22:      else
23:         if first(entity.email) ϵ trecexperts then
24:            merge expert and add doc id
25:            done ← 1
26:         end if
27:      end if
28:   end if
29:
30:   if done = 0 then
31:      if name ≠ 'Unknown' then
32:         if entity.name ϵ trecexperts then
33:            merge expert and add doc id
34:            done = 1
35:         else
36:            if length(entity.name) < 2 then
37:               for all expert ϵ trecexperts do
38:                  if expert.name ∼ entity.name then
39:                     merge expert and add doc id
40:                     done ← 1
41:                  end if
42:               end for
43:            else
44:               for all expert ϵ trecexperts do
45:                  if expert.name ∼ majority(entity.name) then
46:                     merge expert and add doc id
47:                     done ← 1
48:                  end if
```

```
49:                 end for
50:               end if
51:             end if
52:           end if
53:         end if
54: end for
55:
56: for all expert ϵ trecexperts do
57:     print expert.id, expert.names, expert.emails, expert.docs to output file
58: end for
```

The algorithm takes as input a list of name-email pairs for the given candidate experts and a list of name-email-docid triples corresponding to person instances in the corpus. We use absolute and fuzzy matching across names and emails to determine all the instances of each expert in first list, in the second list. When matching an expert with a person instance, the algorithm first checks if the email address is the same and if so it merges them. If not then it checks whether the first part of both emails (the text before the @ character) is the same. If so then again the algorithm merges them. Otherwise, the algorithm tries to match the names if the name of the person instance consists of at least two tokens. We use a minimum of two tokens to reduce the likelihood of ambiguity. If the names are the same then the algorithm merges the expert and the person instance. Otherwise if a majority of the tokens in the expert name are in the instance name then also they are merged. If none of these criteria match then no merging takes place. At each merge step the algorithm adds the docid of the person instance to the list of docids for the expert. The output of this algorithm is a list of names, emails and docids in the corpus for each candidate expert.

REFERENCES

[1] "ACM SIGIR Special Interest Group on Information Retrieval Homepage". http://www.acm.org/sigir/ (accessed Oct. 1, 2007).

[2] "Apache Lucene - Overview". http://lucene.apache.org/java/docs/ (accessed Oct. 1, 2007).

[3] "ClearForest :: From Information To Action". http://www.clearforest.com (accessed Oct. 1, 2007).

[4] "ClearForest SWS >> Tech Specs". http://sws.clearforest.com/ (accessed Oct. 1, 2007).

[5] "Data Mining and Text Mining for Bioinformatics". http://kd.cs.uni-magdeburg.de/ws03.html (accessed Oct. 1, 2007).

[6] "DIP:Home". http://dip.doe-mbi.ucla.edu/ (accessed Oct. 1, 2007).

[7] "Dublin Core Metadata Initiative (DCMI)". http://dublincore.org/ (accessed Oct. 1, 2007).

[8] "ECHELON - Wikipedia, the free encyclopedia". http://en.wikipedia.org/wiki/ECHELON (accessed Oct. 1, 2007).

[9] "Entrez-PubMed". http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=PubMed (accessed Oct. 1, 2007).

[10] "ExPASy - Swiss-Prot and TrEMBL". http://us.expasy.org/sprot/ (accessed Oct. 1, 2007).

[11] "FTP Information - EDGAR Database". http://www.sec.gov/edgar/searchedgar/ftpusers.htm (accessed Oct 1, 2007).

[12] "IW3C2 - Welcome". http://www.iw3c2.org/ (accessed Oct. 1, 2007).

[13] "KDD-2000 Workshop on Text Mining". http://www.cs.cmu.edu/~dunja/WshKDD2000.html (accessed Oct. 1, 2007).

[14] "LingPipe Home". http://www.alias-i.com/lingpipe/ (accessed Oct. 1, 2007).

[15] "LocusLink Introduction". http://www.ncbi.nlm.nih.gov/LocusLink/ (accessed Oct. 1, 2007).

[16] "Main Page - Wikipedia, the free encyclopedia". http://en.wikipedia.org/wiki/Main_Page (accessed Oct. 1, 2007).

[17] "Online Mendelian Inheritance in Man". http://www.ncbi.nlm.nih.gov/omim/ (accessed Oct. 1, 2007).

[18] "Research - Meta". http://meta.wikimedia.org/wiki/Research (accessed Oct. 1, 2007).

[19] "Reuters-21578 Text Categorization Collection". http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html (accessed Oct 1, 2007).

[20] "Text Mining Workshop, April 22, 2006 (Bethesda, MD)". http://www.cs.utk.edu/tmw06/ (accessed Apr. 1, 2006).

[21] "Text REtrieval Conference (TREC) QA Data". http://trec.nist.gov/data/qa.html (accessed Oct. 1, 2007).

[22] "The GDB Human Genome Database". http://www.gdb.org (accessed Oct. 1, 2007).

[23] "The Human Genome Organisation". http://www.gene.ucl.ac.uk/hugo/ (accessed Oct. 1, 2007).

[24] "UCINET 6 Social Network Analysis Software". http://www.analytictech.com/ucinet/ucinet.htm (accessed Oct. 1, 2007).

[25] "www.genatlas.org". http://www.genatlas.org/ (accessed Oct. 1, 2007).

[26] L. Adamic and E. Adar. Friends and Neighbors on the Web. *Social Networks*, 25(3):211–230, 2001.

[27] R. Agrawal, T. Imielinski, and A.N. Swami. Mining Association Rules between Sets of Items in Large Databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216, 1993.

[28] E. Amitay. Hypertext: The importance of being different. Master's thesis, University of Edinburgh, 1997.

[29] A. Aronson. Effective mapping of biomedical text to the UMLS metathesaurus: The MetaMap program. In *Proceedings of the American Medical Informatics Association*, pages 17–21, 2001.

[30] R. Bekkerman and A. McCallum. Disambiguating Web appearances of people in a social network. In *Proceedings of the 14th International World Wide Web Conference (WWW-2005)*, pages 463–470, 2005.

[31] M. Ben-Dov, W. Wu, P. Cairns, and R. Feldman. Improving knowledge discovery by combining text-mining and link analysis techniques. In *Proceedings of the SIAM International Conference on Data Mining*, 2004.

[32] A. Bernstein, S. Clearwater, S. Hill, C. Perlich, and F. Provost. Discovering Knowledge from Relational Data Extracted from Business News. In *Proceedings of the Workshop on Multi-Relational Data Mining*, pages 7–20, 2002.

[33] M.V. Blagosklonny and A.B. Pardee. Conceptual biology: unearthing the gems. *Nature*, 416(6879):373–374, 2002.

[34] C. Blaschke, M.A. Andrade, C. Ouzounis, and A. Valencia. Automatic Extraction of Biological Information from Scientific Text: Protein-Protein Interactions. In *Proceedings of the International Conference on Intelligent Systems for Molecular Biology*, pages 60–67, 1999.

[35] C. Blaschke and A. Valencia. The potential use of SUISEKI as a protein interaction discovery tool. *Genome Informatics*, 12:123–134, 2001.

[36] O. Bodenreider. GenNav. `http://mor.nlm.nih.gov/perl/gennav.pl` (accessed Oct. 1, 2007).

[37] S. Böttcher, L. Werner, and R. Beckmann. Enhanced Information Retrieval by Using HTML Tags. In *Proceedings of The 2005 International Conference on Data Mining (DMIN 2005)*, pages 24–29, 2005.

[38] E. Brill. Some Advances in Transformation-Based Part of Speech Tagging. In *Proceedings of the National Conference on Artificial Intelligence (AAAI-94)*, pages 722–727, 1994.

[39] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[40] C. Buckley and E.M. Voorhees. Retrieval evaluation with incomplete information. In *Proceedings of the 27th ACM SIGIR conference*, pages 25–32, 2004.

[41] D. Buscaldi and P. Rosso. A Bag-of-Words Based Ranking Method for the Wikipedia Question Answering Task. In *Proceedings of the Cross Language Evaluation Forum (CLEF)*, pages 550–553, 2006.

[42] J.P. Callan and T. Mitamura. Knowledge-based extraction of named entities. In *Proceedings of the Conference on Information and Knowledge Management (CIKM-2002)*, pages 532–537, 2002.

[43] J.L. Carlson. Electoral Accountability, Party Loyalty, and Roll-Call Voting in the U.S. Senate. In *Proceedings of the Party Effects in the U.S. Senate Conference (Duke University)*, April 2006.

[44] S. Chakrabarti, B.E. Dom, S.R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, D. Gibson, and J. Kleinberg. Mining the Web's Link Structure. *IEEE Computer*, 32(8):60–67, 1999.

[45] S. Chakrabarti, M. Joshi, and V. Tawde. Enhanced Topic Distillation Using Text, Markup Tags, and Hyperlinks. In *Proceedings of the 24th International ACM SIGIR Conference*, pages 208–216, 2001.

[46] H. Chen. COPLINK. `http://www.coplink.net/` (accessed Oct. 1, 2007).

[47] H. Chen and B.M. Sharp. Content-rich biological network constructed by mining PubMed abstracts. *BMC Bioinformatics*, 5:147, 2004.

[48] J. Clinton, S. Jackman, and D. Rivers. The Statistical Analysis of Roll Call Data. *American Political Science Review*, pages 355–370, May 2004.

[49] J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20:37–46, 1960.

[50] J.G. Conrad and M.H. Utt. A System for Discovering Relationships by Feature Extraction from Text Databases. In *Proceedings of the 17th International ACM SIGIR Conference (Special Issue of the SIGIR Forum)*, pages 260–270, 1994.

[51] F.M. Couto, B. Martins, and M.J. Silva. Classifying Biological Articles Using Web Resources. In *Proceedings of the ACM Symposium on Applied Computing*, pages 111–115, 2004.

[52] N. Craswell, A.P. de Vries, and I. Soboroff. Overview of the TREC 2005 enterprise track. *TREC 2005 Conference Notebook*, 2005.

[53] N. Craswell, D. Hawking, A. Vercoustre, and P. Wilkins. P@NOPTIC Expert: Searching for Experts not just for Documents. In *Poster Proceedings of the 7th Australian World Wide Web Conference*, 2001.

[54] A. Culotta, Ron Bekkerman, and A. McCallum. Extracting Social Networks and Contact Information from Email and the Web. In *Proceedings of the 1st Conference on Email and Spam (CEAS 2004)*, 2004.

[55] M.L. Culter, Y. Shih, and W. Meng. Using the Structure of HTML Documents to Improve Retrieval. In *Proceedings of the USENIX Symposium on Internet Technologies and Systems*, pages 241–252, 1997.

[56] R. Feldman and I. Dagan. Knowledge Discovery in Textual Databases (KDT). In *Knowledge Discovery and Data Mining*, pages 112–117, 1995.

[57] C. G. Figuerola, J. L. A. Berrocal, A. F. Zazo Rodríguez, and E. Rodríguez. Improving Web Pages Retrieval Using Combined Fields. In *Proceedings of the Cross Language Evaluation Forum*, pages 820–825, 2006.

[58] J.R. Finkel, S. Dingare, H. Nguyen, M. Nissim, G. Sinclair, and C. Manning. Exploiting Context for Biomedical Entity Recognition: From Syntax to the Web. In *Proceedings of the International Joint Workshop on NLP in Biomedicine and its Applications (JNLPBA)*, pages 88–91, 2004.

[59] J.R. Finkel, T. Grenager, and C. Manning. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pages 363–370, 2005.

[60] G. Flake, S. Lawrence, and C.L. Giles. Efficient Identification of Web Communities. In *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 150–160, 2000.

[61] B. Franz. Lingua-EN-Tagger. `http://search.cpan.org/dist/Lingua-EN-Tagger/` (accessed Oct. 1, 2007).

[62] B. Franz. Lingua-Stem. `http://search.cpan.org/dist/Lingua-Stem/` (accessed Oct. 1, 2007).

[63] Y. Fu, W. Yu, Y. Li, Y. Liu, M. Zhang, and S. Ma. THUIR at TREC 2005: Enterprise track. In *Proceedings of the Fourteenth Text REtrieval Conference (TREC 2005)*, 2005.

[64] E. Gabrilovich and S. Markovitch. Overcoming the Brittleness Bottleneck using Wikipedia: Enhancing Text Categorization with Encyclopedic Knowledge. In *Proceedings of the 21st National Conference on Artificial Intelligence*, 2006.

[65] N. Glance, M. Hurst, K. Nigam, M. Siegler, R. Stockton, and T. Tomokiyo. Deriving marketing intelligence from online discussion. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2005)*, pages 419–428, 2005.

[66] J. Golbeck and J. Hendler. Inferring Trust Relationships in Web-Based Social Networks. *ACM Transactions on Internet Technology (TOIT)*, 6(4), 2005.

[67] M.D. Gordon, R.K. Lindsay, and W. Fan. Literature-based discovery on the World Wide Web. *ACM Transactions on Internet Technology (TOIT)*, 2(4):261–275, 2002.

[68] J.D. Griffin. Senate Apportionment as a Source of Political Inequality. *Legislative Studies Quarterly*, 31(3):405–432, August 2006.

[69] D.B. Guruge and R.J. Stonier. Intelligent Document Filter for the Internet. volume 3755 of *Lecture Notes in Computer Science*, pages 161–175. Springer Verlag, 2006.

[70] V. Hatzivassiloglou, P. A. Duboué, and A. Rzhetsky. Disambiguating proteins, genes, and RNA in text: a machine learning approach. *Bioinformatics*, 17(1):S97–S106, 2001.

[71] M. Hearst. TextTiling: Segmenting Text into Multi-Paragraph Subtopic Passages. *Computational Linguistics*, 23(1):33–64, 1997.

[72] M. Hearst. Untangling text data mining. In *Proceedings of ACL'99: The 37th Annual Meeting of the Association for Computational Linguistics*, 1999.

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

[74] J. Hopcroft and J. Wong. Linear time algorithm for isomorphism of planar graphs. In *Proceedings of the Sixth Annual ACM Symposium on Theory of Computing*, pages 172–184, 1974.

[75] T.K. Jenssen, A. Laegreid, J. Komorowski, and E. Hovig. A literature network of human genes for high-throughput analysis of gene expression. *Nature Genetics*, 28:21–28, 2001.

[76] T. Joachims, N. Cristianini, and J. Shawe-Taylor. Composite Kernels for Hypertext Categorisation. In *Proceedings of 18th International Conference on Machine Learning*, pages 250–257, 2001.

[77] J. Gerdes Jr. EDGAR-Analyzer: automating the analyses of corporate data contained in the SEC's EDGAR database. *Decision Support Systems*, 35(1):7–29, 2003.

[78] H. Kautz, B. Selman, and M. Shah. Referral Web: Combining Social Networks and Collaborative Filtering. *Communications of the ACM*, 40(3):63–65, 1997.

[79] J. Kim, T. Ohta, Y. Teteisi, and J. Tsujii. GENIA corpus - a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl. 1):i180–i182, 2003.

[80] S. Kim, H. Alani, W. Hall, P.H. Lewis, D.E. Millard, N.R. Shadbolt, and M.J. Weal. Artequakt: Generating Tailored Biographies with Automatically Annotated Fragments from the Web. In *Proceedings of the Workshop on the Semantic Authoring, Annotation & Knowledge Markup*, pages 1–6, 2002.

[81] D. Konopnicki and O. Shmueli. W3QS: A Query System for the World Wide Web. In *Proceedings of the 21st Conference on Very Large Databases*, pages 54–65, 1995.

[82] R. Kosala and H. Blockeel. Web Mining Research: A Survey. *SIGKDD: SIGKDD Explorations: Newsletter of the Special Interest Group (SIG) on Knowledge Discovery & Data Mining*, 2, 2000.

[83] J.H. Kroeze, M.C. Matthee, and T.J.D. Bothma. Differentiating data- and text-mining terminology. In *Proceedings of the 2003 annual research conference of the South African institute of computer scientists and information technologists on Enablement through technology*, pages 93–101, 2003.

[84] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the Web for emerging cyber-communities. In *Proceedings of the 8th International World Wide Web Conference (WWW-1999)*, pages 403–415, 1999.

[85] M. Lapata and K. Frank. The Web as a Baseline: Evaluating the Performance of Unsupervised Web-based Models for a Range of NLP Tasks. In *Proceedings of the HLT/NAACL Conference*, pages 121–128, 2004.

[86] W. Li, R. Srihari, C. Niu, and X. Li. Entity Profile Extraction from Large Corpora. In *Proceedings of the Pacific Association for Computational Linguistics 2003 (PACLING-2003)*, 2003.

[87] A. Lih. Wikipedia as Participatory Journalism: Reliable Sources? Metrics for evaluating collaborative media as a news resource. In *Proceedings of the 5th International Symposium on Online Journalism*, 2004.

[88] B. Liu, C.W. Chin, and H.T. Ng. Mining Topic-Specific Concepts and Definitions on the Web. In *Proceedings of the 12th International World Wide Web conference (WWW-2003)*, pages 251–260, 2003.

[89] S.K. Madria, S.S. Bhowmick, W.K. Ng, and E. Lim. Research Issues in Web Data Mining. In *Proceedings of the Data Warehousing and Knowledge Discovery, First International Conference*, pages 303–312, 1999.

[90] W.H. Majoros, G.M. Subramanian, and M.D. Yandell. Identification of key concepts in biomedical literature using a modified Markov heuristic. *Bioinformatics*, 19(3):402–407, 2003.

[91] B. Malin. Unsupervised Name Disambiguation via Social Network Similarity. In *Proceedings of the Workshop on Link Analysis, Counterterrorism, and Security, at the 2005 SIAM International Conference on Data Mining*, pages 93–102, 2005.

[92] E.M. Marcotte, I. Xenarios, and D. Eisenberg. Mining literature for protein-protein interactions. *Bioinformatics*, 17(4):359–363, April 2001.

[93] Y. Matsuo, J. Mori, M. Hamasaki, K. Ishida, T. Nishimura, H. Takeda, K. Hasida, and M. Ishizuka. POLYPHONET: an advanced social network extraction system from the web. In *Proceedings of the 15th International World Wide Web Conference (WWW-2006)*, pages 397–406, 2006.

[94] G. Mihaila. WebSQL—A SQL-like Query Language for the World Wide Web. Master's thesis, Department of Computer Science, University of Toronto, 1996.

[95] B. Mobasher, R. Cooley, and J. Srivastava. Automatic personalization based on Web usage mining. *Communications of the ACM*, 43(8):142–151, 2000.

[96] A. Morgan, L. Hirschman, A. Yeh, and M. Colosimo. Gene Name Extraction Using FlyBase Resources. In Sophia Ananiadou and Jun'ichi Tsujii, editors, *Proceedings of the ACL 2003 Workshop on Natural Language Processing in Biomedicine*, pages 1–8, 2003.

[97] S. Morgenstern. *Patterns of Legislative Politics : Roll-Call Voting in Latin America and the United States*. Cambridge University Press, 2004.

[98] U.Y. Nahm and R.J. Mooney. A Mutually Beneficial Integration of Data Mining and Information Extraction. In *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI-2000)*, pages 627–632, 2000.

[99] J. Neto, A. Santos, C. Kaestner, and A. Freitas. Document clustering and text summarization. In *Proceedings of the 4th International Conference on Practical Applications of Knowledge Discovery and Data Mining (PADD-2000)*, pages 41–55, 2000.

[100] Michel Neuhaus and Horst Bunke. A Quadratic Programming Approach to the Graph Edit Distance Problem. In *Graph-Based Representations in Pattern Recognition*, volume 4538 of *Lecture Notes in Computer Science*, pages 92–102. Springer, 2007.

[101] D. Newman, C. Chemudugunta, P. Smyth, and M. Steyvers. Analyzing Entities and Topics in News Articles Using Statistical Topic Models. In *IEEE International Conference on Intelligence and Security Informatics*, pages 93–104, 2006.

[102] J.C. Park, H.S. Kim, and J.J. Kim. Bidirectional incremental parsing for automatic pathway identification with combinatory categorical grammar. In *Proceedings of the Pacific Symposium on Biocomputing*, volume 6, pages 396–407, 2001.

[103] G. Petasis, V. Karkaletsis, C. Grover, B. Hachey, M. T. Pazienza, M. Vindigni, and J. Coch. Adaptive, Multilingual Named Entity Recognition in Web Pages. In *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI-2004)*, pages 1073–1074, 2004.

[104] J.M. Ponte and W.B. Croft. A Language Modeling Approach to Information Retrieval. In *Research and Development in Information Retrieval*, pages 275–281, 1998.

[105] M. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[106] H. Raghavan, J. Allan, and A. McCallum. An Exploration of Entity Models, Collective Classification and Relation Description. In *Proceedings of KDD Workshop on Link Analysis and Group Detection (LinkKDD-2004)*, pages 1–10, 2004.

[107] D. Ravichandran and E. Hovy. Learning surface text patterns for a Question Answering system. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 41–47, 2001.

[108] J.C. Reynar and A. Ratnaparkhi. A Maximum Entropy Approach to Identifying Sentence Boundaries. In *Proceedings of the 5th Conference on Applied Natural Language Processing*, 1997.

[109] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

[110] T. Scheffer, C. Decomain, and S. Wrobel. Mining the Web with Active Hidden Markov Models. In *Proceedings of the IEEE International Conference on Data Mining (ICDM-2001)*, pages 645–646, 2001.

[111] A.K. Sehgal, X.Y. Qiu, and P. Srinivasan. Mining MEDLINE Metadata to Explore Genes and their Connections. In *Proceedings of the SIGIR 2003 Workshop on Text Analysis and Search for Bioinformatics*, 2003.

[112] A.K. Sehgal and P. Srinivasan. Manjal: a text mining system for medline. In *Poster Proceedings of the 28th ACM SIGIR conference*, pages 680–680, 2005.

[113] A.K. Sehgal and P. Srinivasan. Retrieval with gene queries. *BMC Bioinformatics*, 7(1):220, 2006.

[114] A.K. Sehgal and P. Srinivasan. Profiling Topics on the Web. pages 1–8, 2007.

[115] T. Sekimizu, H.S. Park, and J. Tsujii. Identifying the interactions between Genes and Gene products based on frequently seen verbs in MEDLINE abstracts. In S. Miyano and T. Takagi, editors, *Genome Informatics (GIW' 98)*, pages 62–71. Universal Academy Press, 1998.

[116] H. Shatkay, S. Edwards, W. J. Wilbur, and M. Boguski. Genes, Themes and Microarrays. Using information retrieval for large-scale gene analysis. In *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology*, pages 317–328, 2000.

[117] I. Soboroff, A.P. de Vries, and N. Craswell. Overview of the TREC 2006 enterprise track. *TREC 2006 Conference Notebook*, 2006.

[118] S. Soderland. Learning to Extract Text-Based Information from the World Wide Web. In *Proceedings of the 3rd International Conference on Knowledge Discovery and DataMining*, pages 251–254, 1997.

[119] P. Srinivasan. MeSHmap: A text mining tool for MEDLINE. In *Proceedings of the American Medical Informatics Annual Symposium (AMIA)*, pages 642–646, 2001.

[120] P. Srinivasan, B. Libbus, and A.K. Sehgal. Mining MEDLINE: Postulating a Beneficial Role for Curcumin Longa in Retinal Diseases. In *Proceedings of the HLT-NAACL 2004 Workshop: BioLink 2004, Linking Biological Literature, Ontologies and Databases*, pages 33–40, 2004.

[121] P. Srinivasan and A.K. Sehgal. Mining MEDLINE for Similar Genes and Similar Drugs, 2003. Technical Report: Department of Computer Science, The University of Iowa.

[122] M. Stephens, M. Palakal, S. Mukhopadhyay, R. Raje, and J. Mostafa. Detecting Gene Relations from MEDLINE Abstracts. In *Proceedings of the Sixth Annual Pacific Symposium on Biocomputing (PSB)*, 2001.

[123] K. Sugiyama, K. Hatano, M. Yoshikawa, and S. Uemura. Extracting Information on Protein-Protein Interactions. *Genome Informatics*, 14:699–700, 2003.

[124] A. Sun, E. Lim, and W. Ng. Web classification using support vector machine. In *Proceedings of the fourth international workshop on Web information and data management*, pages 96–99, 2002.

[125] D.R. Swanson. Fish oil, Raynaud's syndrome, and undiscovered public knowledge. *Perspectives in Biology and Medicine*, 30(1):7–18, 1986.

[126] D.R. Swanson. Undiscovered public knowledge. *Library Quarterly*, 56:103–118, 1986.

[127] D.R. Swanson. Two medical literatures that are logically but not bibliographically connected. *Journal of the American Society for Information Science*, 38(4):228–233, 1987.

[128] D.R. Swanson. Migraine and Magnesium: Eleven neglected connections. *Perspectives in Biology and Medicine*, 31:526–557, 1988.

[129] D.R. Swanson and N.R. Smalheiser. Indomethacin and Alzheimer's disease. *Neurology*, 46:583, 1996.

[130] D.R. Swanson and N.R. Smalheiser. Linking estrogen to Alzheimer's disease: An informatics approach. *Neurology*, 47:809–810, 1996.

[131] D.R. Swanson and N.R. Smalheiser. Calcium-independent phospholipase A2 and Schizophrenia. *Archives of General Psychiatry*, 55(8):752–753, 1998.

[132] D.R. Swanson, N.R. Smalheiser, and A. Bookstein. Information Discovery from Complementary Literatures: Categorizing Viruses as Potential Weapons. *Journal of the American Society for Information Science And Technology*, 52(10):797–812, 2001.

[133] A. Tan. Text mining: The state of the art and the challenges. In *Proceedings of the Pacific Asia Conf on Knowledge Discovery and Data Mining PAKDD'99 workshop on Knowledge Discovery from Advanced Databases.*, pages 65–70, 1999.

[134] P. Tan and V. Kumar. Mining Indirect Associations in Web Data. In *Proceedings of WEBKDD 2001 - Mining Web Log Data Across All Customers Touch Points*, pages 145–166, 2001.

[135] J. Thomas, D. Milward, C. Ouzounis, S. Pulman, and M. Carroll. Automatic Extraction of Protein Interactions from Scientific Abstracts. In *Proceedings of the Pacific Symposium on Biocomputing*, volume 5, pages 538–549, 2000.

[136] L. Vijjappu, A. Tan, and C. Tan. Web Structure Analysis for Information Mining. In A. Antonacopoulos and J. Hu, editors, *Web Document Analysis Challenges and Opportunities*. World Scientific, 2003.

[137] J. Voss. Measuring Wikipedia. In *Proceedings of the 10th International Conference of the International Society for Scientometrics and Informetrics*, 2005.

[138] X. Wang, N. Mohanty, and A. McCallum. Group and Topic Discovery from Relations and Their Attributes. In *Proceedings of the Workshop on Link Analysis: Dynamics and Static of Large Networks (LinkKDD)*, pages 28–35, 2005.

[139] M. Weeber, H. Klein, L.T.W. de Jong-van den Berg, and R. Vos. Using Concepts in Literature-Based Discovery: Simulating Swanson's Raynaud-Fish Oil and Migraine-Magnesium Discoveries. *Journal of the American Society for Information Science And Technology*, 52(7):548–557, 2001.

[140] M. Weeber, B.J.A. Schijvenaars, E.M. van Mulligen, B. Mons, R. Jelier, C. van der Eijk, and J.A. Kors. Ambiguity of Human Gene Symbols in LocusLink and MEDLINE: Creating an Inventory and a Disambiguation Test Collection. In *Proceedings of the American Medical Informatics Association (AMIA) Symposium*, pages 704–708, 2003.

[141] T. Yan, M. Jacobsen, H. Garcia-Molina, and U. Dayal. From user access patterns to dynamic hypertext linking. In *Proceedings of the 5th International World Wide Web Conference on Computer networks and ISDN systems*, pages 1007–1014, 1996.

[142] Z. Yao and B. Choi. Bidirectional Hierarchical Clustering for Web Mining. In *Proceedings of the IEEE/WIC International on Web Intelligence (WI-2003)*, pages 620–624, 2003.

[143] A.C. Yeo, K.A. Smith, R.J. Willis, and M. Brooks. Clustering technique for risk classification and prediction of claim cost in the automobile insurance industry. *International Journal of Intelligent Systems in Accounting, Finance and Management*, pages 39–50, 2001.

[144] L. Yi, B. Liu, and X. Li. Eliminating noisy information in Web pages for data mining. In *Knowledge Discovery and Data Mining*, pages 296–305, 2003.

[145] H. Yu and E. Agichtein. Extracting synonymous gene and protein terms from biological literature. *Bioinformatics*, 10(1):340–349, 2003.

[146] J. Zhu, D. Song, S. Rüger, M. Eisenstadt, and E. Motta. The Open University at TREC 2006 Enterprise Track Expert Search Task. In *Proceedings of the Fifteenth Text REtrieval Conference (TREC 2006)*, 2006.