CSI IN THE WEB 2.0 AGE: DATA COLLECTION, SELECTION, AND INVESTIGATION FOR KNOWLEDGE DISCOVERY

by

Tianjun Fu

A Dissertation Submitted to the Faculty of the
Committee On Business Administration
In Partial Fulfillment of the Requirements
For the Degree of
DOCTOR OF PHILOSOPHY
WITH A MAJOR IN MANAGEMENT
In the Graduate College
THE UNIVERSITY OF ARIZONA

2011
As members of the Dissertation Committee, we certify that we have read the dissertation prepared by Tianjun Fu entitled CSI in the Web 2.0 Age: Data Collection, Selection, and Investigation for Knowledge Discovery and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy.

Date: 01/06/2012
Daniel Zeng

Date: 01/06/2012
Paulo Goes

Date: 01/06/2012
Zhu Zhang

Final approval and acceptance of this dissertation is contingent upon the candidate’s submission of the final copies of the dissertation to the Graduate College.

I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

Date: 01/06/2012
Dissertation Director: Daniel Zeng
STATEMENT BY AUTHOR

This dissertation has been submitted in partial fulfillment of requirements for an advanced degree at the University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this dissertation are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: Tianjun Fu
I would like to thank my advisor, Dr. Daniel Zeng for his support, trust, guidance, and encouragement along the way. What I have learned from him both academically and personally will significant benefit my future career. I am grateful to my dissertation committee members, Dr. Paulo Goes, Dr. Zhu Zhang, and my minor committee member, Dr. Sandiway Fong, for their support and kindness. Special thanks to Dr. Hsinchun Chen for his help and guidance on several dissertation chapters when I worked in his AI lab. I also thank the rest of MIS faculty members and staffs for their support during my studies.

I would also like to thank my colleagues and friends, including Ahmed Abbasi, Xin Li, Daning Hu, Wei Chang, Runpu Sun, Hsinmin Lu, Nichalin Summerfield, Chunneng Huang, Li Fan, Shuo Zeng, Ximing Yu, Yulei Zhang, Yan Dang, Shaokun Fan, Xinlei Zhao, Zheren Hu, Ping Yan, Ying Liu, Manlu Liu, Yilu Zhou, Jialun Qin, Gang Wang, Jiannan Wang, Jiexun Li, Gang Wang, Siddharth Kaza, Rob Schumaker, Ted Elhourani, Yida Chen, David Zimbra, Sven Thomas, and Katherine Carl for their companionship. I have worked closely with Ahmed, Xin, Chunneng, and David in my research and I greatly appreciate their support.

Most of all, I am deeply grateful to my parents, Renji Fu and Jinlan Liu, and my wife, Jinwei Zhang. It would have been impossible to write this dissertation without their constant and unconditional support.
DEDICATION

This dissertation is dedicated to my parents and my wife

for their unconditional love and support.
<table>
<thead>
<tr>
<th>TABLE OF CONTENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF ILLUSTRATIONS .................................................................</td>
</tr>
<tr>
<td>LIST OF TABLES ..................................................................................</td>
</tr>
<tr>
<td>ABSTRACT ...........................................................................................</td>
</tr>
<tr>
<td>CHAPTER 1: INTRODUCTION ...............................................................</td>
</tr>
<tr>
<td>1.1 Motivation and Objectives .............................................................</td>
</tr>
<tr>
<td>1.2 Dissertation Framework ...............................................................</td>
</tr>
<tr>
<td>CHAPTER 2: A FOCUSED CRAWLER FOR DARK WEB FORUMS ................</td>
</tr>
<tr>
<td>2.1 Introduction ..................................................................................</td>
</tr>
<tr>
<td>2.2 Related Work: Focused and Hidden Web Crawlers .........................</td>
</tr>
<tr>
<td>2.2.1 Accessibility ............................................................................</td>
</tr>
<tr>
<td>2.2.2 Collection Type .........................................................................</td>
</tr>
<tr>
<td>2.2.3 Content Richness ......................................................................</td>
</tr>
<tr>
<td>2.2.4 URL Ordering Features .............................................................</td>
</tr>
<tr>
<td>2.2.5 URL Ordering Techniques ..........................................................</td>
</tr>
<tr>
<td>2.2.6 Collection Update Procedure .....................................................</td>
</tr>
<tr>
<td>2.2.7 Summary of Previous Research ..................................................</td>
</tr>
<tr>
<td>2.3 Research Gaps and Questions .......................................................</td>
</tr>
<tr>
<td>2.3.1 Focused Crawling of the Hidden Web ..........................................</td>
</tr>
<tr>
<td>2.3.2 Content Richness ......................................................................</td>
</tr>
<tr>
<td>2.3.3 Collection Recall Improvement ..................................................</td>
</tr>
<tr>
<td>2.3.4 Web Forum Collection-Update Strategies .....................................</td>
</tr>
<tr>
<td>2.3.5 Research Questions ...................................................................</td>
</tr>
<tr>
<td>2.4 Research Design ...........................................................................</td>
</tr>
<tr>
<td>2.4.1 Proposed Dark Web Forum Crawling System ................................</td>
</tr>
<tr>
<td>2.4.2 Accessibility ............................................................................</td>
</tr>
<tr>
<td>2.4.3 Recall-Improvement Mechanism ................................................</td>
</tr>
<tr>
<td>2.4.4 Incremental Crawling for Collection Updating ............................</td>
</tr>
<tr>
<td>2.5 System Design .............................................................................</td>
</tr>
<tr>
<td>2.5.1 Forum Identification ..................................................................</td>
</tr>
<tr>
<td>2.5.2 Forum Preprocessing ..................................................................</td>
</tr>
<tr>
<td>2.5.2.1 Forum Accessibility .................................................................</td>
</tr>
<tr>
<td>2.5.2.2 Forum Structure ......................................................................</td>
</tr>
<tr>
<td>2.5.2.3 Wrapper Generation ...............................................................</td>
</tr>
<tr>
<td>2.5.3 Forum Spidering ..........................................................................</td>
</tr>
<tr>
<td>2.5.4 Forum Storage and Analysis ......................................................</td>
</tr>
<tr>
<td>2.5.4.1 Statistics Generation ..............................................................</td>
</tr>
<tr>
<td>2.5.4.2 Duplicate Multimedia Removal ...............................................</td>
</tr>
<tr>
<td>2.5.5 Dark Web Forum Crawling System Interface ..................................</td>
</tr>
<tr>
<td>2.6 Evaluation ..................................................................................</td>
</tr>
<tr>
<td>2.6.1 Forum Accessibility Experiment ...............................................</td>
</tr>
<tr>
<td>2.6.2 Spidering Parameter Experiment .................................................</td>
</tr>
<tr>
<td>2.6.3 Forum Collection-update Experiment .........................................</td>
</tr>
<tr>
<td>Section</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
</tr>
<tr>
<td>2.6.4 Forum Collection Statistics</td>
</tr>
<tr>
<td>2.7 Dark Web Forum Case Study</td>
</tr>
<tr>
<td>2.7.1 Topical Analysis</td>
</tr>
<tr>
<td>2.7.2 Interaction Analysis</td>
</tr>
<tr>
<td>2.8 Conclusions and Future Directions</td>
</tr>
<tr>
<td>CHAPTER 3: SENTIMENTAL SPIDERING: LEVERAGING OPINION</td>
</tr>
<tr>
<td>3.1 Introduction</td>
</tr>
<tr>
<td>3.2 Literature Review</td>
</tr>
<tr>
<td>3.3 Research Design</td>
</tr>
<tr>
<td>3.3.1 Text Classifier Module</td>
</tr>
<tr>
<td>3.3.2 Graph Comparison Module</td>
</tr>
<tr>
<td>3.4 Evaluation</td>
</tr>
<tr>
<td>3.4.1 Test Bed and Training Data</td>
</tr>
<tr>
<td>3.4.2 Experimental Setup</td>
</tr>
<tr>
<td>3.4.3 Experimental Results</td>
</tr>
<tr>
<td>3.4.4 Impact of Sentiment Information and Graph-based Tunneling</td>
</tr>
<tr>
<td>3.5 Conclusions</td>
</tr>
<tr>
<td>CHAPTER 4: EXPLORING GRAPH-BASED TUNNELING FOR FOCUSED CRAWLERS</td>
</tr>
<tr>
<td>4.1 Introduction</td>
</tr>
<tr>
<td>4.2 Literature Review</td>
</tr>
<tr>
<td>4.2.1 Graph Kernels</td>
</tr>
<tr>
<td>4.2.2 Graph-based Tunneling for Focused Crawler</td>
</tr>
<tr>
<td>4.3 Research Design</td>
</tr>
<tr>
<td>4.4 Evaluation</td>
</tr>
<tr>
<td>4.5 Conclusion</td>
</tr>
<tr>
<td>CHAPTER 5: TEXT-BASED VIDEO CONTENT CLASSIFICATION FOR ONLINE VIDEO-SHARING SITES</td>
</tr>
<tr>
<td>5.1 Introduction</td>
</tr>
<tr>
<td>5.2 Literature Review</td>
</tr>
<tr>
<td>5.2.1 Video Domains</td>
</tr>
<tr>
<td>5.2.2 Feature Types</td>
</tr>
<tr>
<td>5.2.2.1 Non-Text Features</td>
</tr>
<tr>
<td>5.2.2.2 Text Features</td>
</tr>
<tr>
<td>5.2.2.3 Classification Techniques</td>
</tr>
<tr>
<td>5.3 Research Gaps and Research Questions</td>
</tr>
<tr>
<td>5.4 System Design</td>
</tr>
<tr>
<td>5.4.1 Data Collection</td>
</tr>
<tr>
<td>5.4.2 Feature Generation</td>
</tr>
<tr>
<td>5.4.2.1 Feature Extraction</td>
</tr>
<tr>
<td>5.4.2.2 Feature Sets Generation</td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS - Continued

5.4.2.3 Feature Selection ......................................................................................142
5.4.3 Classification and Evaluation .........................................................................144
5.5 Testbed and Hypotheses .........................................................................................145
5.5.1 Testbed ............................................................................................................145
5.5.2 Hypotheses ......................................................................................................146
5.6 Experiment Results and Discussions ......................................................................147
5.6.1 Comparison of Feature Types .........................................................................149
5.6.2 Comparison of Classification Techniques ......................................................150
5.6.3 Key Feature Analysis ......................................................................................151
5.7 Evaluating the Impact of Video Classification on Social Network Analysis ......155
5.8 Conclusions and Future Directions .................................................................158

CHAPTER 6: A HYBRID APPROACH TO WEB FORUM INTERACTIONAL
COHERENCE ANALYSIS .............................................................................................160
6.1 Introduction ............................................................................................................160
6.2 Related Work .........................................................................................................163
6.2.1 Obstacles to CMC Interactional Coherence ........................................................164
6.2.2 CMC Interactional Coherence Analysis ............................................................166
6.2.2.1 CMC Interactional Coherence Domains ..................................................168
6.2.2.2 CMC Interactional Coherence Research Features ........................................169
6.2.2.2.1 CMC System Features ......................................................................170
6.2.2.2.2 Linguistic Features ............................................................................172
6.2.2.3 Noise Issues in ICA .................................................................................174
6.2.2.4 CMC Interactional Coherence Analysis Techniques ....................................175
6.3 Research Gaps and Questions ................................................................................177
6.4 System Design: Hybrid Interactional Coherence System ......................................178
6.4.1 Data Preparation..............................................................................................180
6.4.2 HIC Algorithm: System Feature Match ..........................................................181
6.4.2.1 Header Information Match .......................................................................181
6.4.2.2 Quotation Match ......................................................................................181
6.4.3 HIC Algorithm: Linguistic Feature Match .......................................................182
6.4.3.1 Direct Address Match ..............................................................................182
6.4.3.2 Lexical Relations: The Lexical Match Algorithm ...................................184
6.4.4 HIC Algorithm: Residual Match .....................................................................186
6.5 Evaluation ..............................................................................................................187
6.5.1 Test Bed ..........................................................................................................187
6.5.2 Experiment 1: Comparison of Techniques ......................................................190
6.5.2.1 Experiment Setup .....................................................................................190
6.5.2.2 Hypotheses ...............................................................................................191
6.5.2.3 Experimental Results ...............................................................................192
6.5.2.4 Hypotheses Results ..................................................................................192
6.5.2.5 Results Discussion ...................................................................................193
6.5.3 Experiment 2: Impact of Noise .......................................................................194
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.5.3.1 Experiment Setup</td>
<td>194</td>
</tr>
<tr>
<td>6.5.3.2 Hypothesis</td>
<td>194</td>
</tr>
<tr>
<td>6.5.3.3 Experimental Results</td>
<td>195</td>
</tr>
<tr>
<td>6.5.3.4 Hypothesis Results</td>
<td>195</td>
</tr>
<tr>
<td>6.5.3.5 Results Discussion</td>
<td>196</td>
</tr>
<tr>
<td>6.5.4 Evaluating the Impact of Interaction Representation: An Example</td>
<td>197</td>
</tr>
<tr>
<td>6.5 Conclusion</td>
<td>199</td>
</tr>
<tr>
<td>CHAPTER 7: CONCLUSIONS AND FUTURE DIRECTIONS</td>
<td>201</td>
</tr>
<tr>
<td>7.1 Conclusions</td>
<td>201</td>
</tr>
<tr>
<td>7.2 Future Directions</td>
<td>204</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>206</td>
</tr>
</tbody>
</table>
LIST OF ILLUSTRATIONS

Figure 1.1: Dissertation Framework .................................................................16
Figure 2.1: Dark Web Forum Crawling System Design .............................................37
Figure 2.2: Dark Web Forum Identification Process ....................................................39
Figure 2.3: Proxies Used for Dark Web Forum Crawling .............................................41
Figure 2.4: URL Traversal Strategies .......................................................................44
Figure 2.5: Spidering Process ....................................................................................45
Figure 2.6: Example Log and Parsed Log Entries .......................................................46
Figure 2.7: Dark Web Forum Crawling System Interface .............................................50
Figure 2.8: Recall Results for Different Settings of Number of Spiders and Proxies per Spider .................................................................54
Figure 2.9: Number of Uncollected Pages for Different Numbers of Spiders .........55
Figure 2.10: Recall Results for Different Settings of Batch Size and Timeout Interval ...56
Figure 2.11: Number of Web Pages in Test Bed across 3 Months/Iterations .............58
Figure 2.12: Results by Iteration for Various Collection Update Procedures ..............59
Figure 2.13: Topical MDS Projections for Domestic Supremacist Forum Authors ....63
Figure 2.14: Author Interaction Network for Domestic Supremacist Forums ............65
Figure 3.1: Example Path for Tunneling .................................................................70
Figure 3.2: System Design for Graph-based Sentiment Crawler ...............................77
Figure 3.3: Illustration of Text Classifier used by GBS Crawler ...............................80
Figure 3.4: Random Walk Paths on a Labelled Web Graph of Page S .......................85
Figure 3.5: Test Bed Statistics by Level .....................................................................91
Figure 3.6: F-Measure Trend for GBS and Comparison Methods ..............................97
Figure 3.7: Precision and Recall Trends for GBS and Comparison Methods ..........99
Figure 3.8: Recall Trends for Pages at Levels 1-6 ....................................................101
Figure 3.9: F-Measure, Recall, and Precision Trends for GBS, GBS-T, and GBS-TS ...104
Figure 4.1: Directed Graphs Mapped to the Same Point in Walks Feature Space .......111
Figure 4.2: Undirected Graphs Mapped to the Same Point in Walks Feature Space ......111
Figure 4.3: F-Measure, Recall, and Precision Trends for RandomWalk, Subtree_D09, Subtree_D09, and CGM .................................................................117
Figure 5.1: Text-based Video Classification System .................................................138
Figure 5.2: Video Classification Accuracies for Different Features and Techniques ....148
Figure 5.3: Key Feature Distribution across User-generated Data Types ....................152
Figure 5.4: Percentage of Key Features by User-generated Data Types ....................152
Figure 5.5: Key Feature Distribution across Feature Types .......................................154
Figure 5.6: Percentage of Key Features by Feature Types .......................................155
Figure 5.7: Social networks of White Supremacy Groups on YouTube .....................157
Figure 6.1: Example of Disrupted Adjacency .........................................................165
Figure 6.2: Features’ Relative Explicit/Implicit Properties .......................................170
Figure 6.3: HIC System Design .............................................................................180
Figure 6.4: Experiment 1 F-measure Performance for each Thread .........................193
Figure 6.5: Experiment 2 F-measure Performance for Each Thread .........................196
Figure 6.6: Social Network Structure of Users in Thread #1 ...................................199
LIST OF TABLES

Table 2.1: Selected Previous Research on Focused Crawling ...........................................30
Table 2.2: Dark Web Forum Accessibility Statistics ..........................................................52
Table 2.3: Macro-Level Results for Different Update Procedures ....................................59
Table 2.4: Dark Web Forum Collection Statistics ..............................................................60
Table 2.5: Dark Web Forum Collection File Statistics .......................................................61
Table 2.6: Domestic Supremacist Forum Test Bed ...........................................................62
Table 2.7: Description of Major Discussion Topics in Test Bed Forums .........................64
Table 3.1: Top 20 RWPs based on Graph Module Training .............................................93
Table 3.2: Standardized Area Under the Curve (AUC) Values .......................................100
Table 4.1: Standardized Area Under the Curve (AUC) Values for First 200k Pages ....118
Table 5.1: Taxonomy of Video Classification Studies ....................................................126
Table 5.2: Selected Major Studies of Video Classification .............................................136
Table 5.3: Text Features Adopted ....................................................................................141
Table 5.4: Feature Counts of Experiment Feature Sets ...................................................147
Table 5.5: Accuracy for Different Feature Sets and Different Techniques .........................149
Table 5.6: Pairwise T-testing of Hypotheses Group 1 (H1) on Accuracy .......................149
Table 5.7: Pairwise T-testing of Hypotheses Group 2 (H2) on Accuracy .......................151
Table 5.8: Degree and Centrality Measures of User “barbituraSS” .................................157
Table 6.1: A Taxonomy of CMC Interactional Coherence Research ..............................167
Table 6.2: Previous CMC Interactional Coherence Studies ............................................167
Table 6.3: Details for Data Sets in Test Bed ....................................................................188
Table 6.4: Interaction Feature Breakdowns across Threads ............................................190
Table 6.5: Experimental Results for Experiment 1 ..........................................................192
Table 6.6: P-values for Pair-wise t-tests on Accuracy for Experiment 1 .........................193
Table 6.7: Experimental Results for Experiment 2 ..........................................................195
Table 6.8: P-values for Pair-wise t-tests on Accuracy for Experiment 2 .........................195
Table 6.9: Degree and Centrality Measures of User “krebsnet” ......................................197
ABSTRACT

The growing popularity of various Web 2.0 media has created massive amounts of user-generated content such as online reviews, blog articles, shared videos, forums threads, and wiki pages. Such content provides insights into web users’ preferences and opinions, online communities, knowledge generation, etc., and presents opportunities for many knowledge discovery problems. However, several challenges need to be addressed: data collection procedure has to deal with unique characteristics and structures of various Web 2.0 media; advanced data selection methods are required to identify data relevant to specific knowledge discovery problems; interactions between Web 2.0 users which are often embedded in user-generated content also need effective methods to identify, model, and analyze.

In this dissertation, I intend to address the above challenges and aim at three types of knowledge discovery tasks: (data) collection, selection, and investigation. Organized in this “CSI” framework, five studies which explore and propose solutions to these tasks for particular Web 2.0 media are presented. In Chapter 2, I study focused and hidden Web crawlers and propose a novel crawling system for Dark Web forums by addressing several unique issues to hidden web data collection. In Chapter 3 I explore the usage of both topical and sentiment information in web crawling. This information is also used to label nodes in web graphs that are employed by a graph-based tunneling mechanism to improve collection recall. Chapter 4 further extends the work in Chapter 3 by exploring the possibilities for other graph comparison techniques to be used in tunneling for focused crawlers. A subtree-based tunneling method which can scale up to large graphs is
proposed and evaluated. Chapter 5 examines the usefulness of user-generated content in online video classification. Three types of text features are extracted from the collected user-generated content and utilized by several feature-based classification techniques to demonstrate the effectiveness of the proposed text-based video classification framework. Chapter 6 presents an algorithm to identify forum user interactions and shows how they can be used for knowledge discovery. The algorithm utilizes a bevy of system and linguistic features and adopts several similarity-based methods to account for interactional idiosyncrasies.
1.1 Motivation and Objectives

Modern information technologies in the Web 2.0 age have created massive amounts of user-generated content such as online reviews, blog articles, shared videos, forums threads, and wiki pages. Such content reflects two key characteristics of Web 2.0, which are architecture of participation (O’Reilly, 2005; Barsky and Purdon, 2006) and wisdom of crowds (Surowiecki, 2004), and provides insights into web users’ preferences and opinions, online communities, information diffusion, and knowledge generation. Consequently the proliferation of user-generated content presents opportunities for knowledge discovery, which is the process of extracting implicit, unknown, and potentially useful information from data (Fayyad et al., 1996), in various domains such as business intelligence, marketing intelligence, and security informatics (Chen, 2009). In order to fully benefit from the massive amounts of user-generated content, several challenges need to be addressed:

- First, data collection procedure has to deal with unique characteristics and structures of various Web 2.0 media. For example, content is mainly organized in thread-message structure in web forums, post-comment structure in blogs, video-comment structure in video-sharing sites, and article-discussion-history structure in wikis. Moreover, blogs provide features such as trackback and blogroll to link relevant bloggers and posts while video sharing sites like YouTube build videos connections by recommending relevant videos and allowing users to make video responses.
Different data collection strategies need to be developed to explore these content structures and Web 2.0 media features.

- Second, advanced data selection methods are required to identify data relevant to specific knowledge discovery problems from the massive amount of data available in the World Wide Web (WWW) or from collected data sets. Besides traditional topic information, user-generated content also contains sentiments, affects, and opinions information of web users. Affects is very important in influencing people’s perceptions and decision making (Picard, 1997). Sentiments and affects also play an important role for analysis of Web 2.0 since content on Web 2.0 media is often more pervasive than topical content (Subasic and Huettner, 2001; Nigam and Hurst, 2004). Therefore such non-topic information needs to be taken into account by both data collection and selection methods for many knowledge discovery problems.

- Finally, interactions between Web 2.0 users are often embedded in user-generated content by either Web 2.0 media or users themselves. User interactions represent one of the fundamental building block metrics for analyzing cyber communities (Fiore et al., 2002; Barcellini et al., 2005) and are invaluable for understanding knowledge flow online (Osterlund and Carlile, 2005; Wasko and Faraj, 2005). The increasing importance of interaction information in the investigation of many knowledge discovery problems necessitates effective methods to identify and model user interactions in Web 2.0.

This dissertation has been motivated by the above challenges and opportunities. The objective is to utilize and enhance current crawling methods, data mining methods, social
network analysis, and graph theories to address the abovementioned knowledge discovery problems in Web 2.0, especially with interaction related data. We organize our studies in a CSI framework (data Collection, Selection, and Investigation) described in the following section.

1.2 Dissertation Framework

![Figure 1.1: Dissertation Framework](image)

The CSI framework focuses on data in the WWW with particular interests in Web 2.0 media and user-generated content, and targets at improving human’s knowledge acquisition and decision making process. The CSI framework specifically aims at the following three types of knowledge discovery tasks:
• Data Collection: to advance the collection of data on Web 2.0, especially user-generated content, by developing efficient and user-friendly focused crawler algorithms.

• Data Selection: to improve the selection, extraction, and representation of relevant information from collected data, especially those that contain interaction information.

• Data Investigation: to identify interactions from selected data and using them to address specific knowledge discovery problems in Web 2.0.

The data collection task emphasizes on where data of interest is located on the WWW and how to collect them efficiently. Chapter 2 proposes a novel crawling system designed to collect content from Dark Web forum (i.e. extremism group forum). The system uses lexicons, search engines, government reports, research centers, and link analysis approach to identify Dark Web forums. A human-assisted access approach was then used to gain access to those forums. Several URL ordering features and techniques enable efficient extraction of forum postings. The system also includes an incremental crawler coupled with a recall-improvement mechanism intended to facilitate enhanced retrieval and updating of collected content. The human-assisted approach significantly improved access to Dark Web forums while the incremental crawler with recall improvement also outperformed standard periodic- and incremental-update approaches.

While Chapter 2 deals with the content structure of Web forums, Chapter 3 explores sentiment information and labeled web graph in web crawling. Chapter 3 proposes a novel focused crawler that incorporates topic and sentiment information as well as a
graph-based tunneling mechanism for enhanced collection of opinion-rich web content regarding a particular topic. The graph-based sentiment crawler (GBS) uses a text classifier that employs both topic and sentiment categorization modules to assess the relevance of candidate pages. This information is also used to label nodes in web graphs that are employed by the tunneling mechanism to improve collection recall by calculating web graph similarities. Experimental results on a test bed encompassing over half a million web pages revealed that GBS was able to provide better precision and recall than six comparison focused crawlers. Moreover, GBS was able to collect a large proportion of the relevant content after traversing far fewer pages than comparison methods.

Chapter 4 further extends the graph comparison work in Chapter 3 and explores the use of other graph comparison techniques in tunneling for focused crawlers in order to find methods that can scale up to large graphs and run fast. Based on the literature review on state-of-the-art graph comparison algorithms, we propose to use subtree-based tunneling methods. In a preliminary experiment, a simple binary subtree based algorithm was evaluated and the results revealed that subtree methods are good at large graphs and run very fast in training. However, several parameters related to subtree patterns such as number of children per node, total number of node, maximum height, and a decay factor need to be tuned in order to reduce the differences in F-measure, precision, and recall between subtree tunneling methods and the random walk one.

The data selection task focuses on how to select, extract, and represent collected data. Chapter 5 represents my work for this task. It proposes a text-based framework for video content classification of online-video sharing Web sites. Different types of user-generated
data, such as video titles, descriptions, and comments, were collected using the APIs provided by online-video sharing sites and used as proxies for online videos. Three types of text features, which are lexical, syntactic, and content-specific features, were extracted to represent the collected user-generated content. Several feature-based classification techniques, including C4.5, Naïve Bayes, and Support Vector Machine, were then used to classify videos. Experimental results showed that the proposed approach was able to classify online videos based on users’ interests with accuracy rates up to 87.2%, and all three types of text features contributed to discriminating videos.

The data investigation task aims at how to accurately identify user interactions and how to use interactions for knowledge discoveries like understanding cyber community development. Chapter 6 explores this task by proposing the Hybrid Interactional Coherence (HIC) algorithm for identification of web forum interaction. HIC utilizes a bevy of system and linguistic features, including message header information, quotations, direct address, and lexical relations. Furthermore, several similarity-based methods including a Lexical Match Algorithm and a sliding window method are utilized to account for interactional idiosyncrasies. Experiments results on two web forums revealed that the proposed HIC algorithm significantly outperformed comparison techniques in terms of precision, recall, and F-measure at both the forum and thread levels. Additionally, an example was used to illustrate how the improved ICA results can facilitate enhanced social network and role analysis capabilities.

Chapter 7 summarizes the dissertation’s contributions to knowledge discovery and information systems, and presents future extensions of this work.
CHAPTER 2: A FOCUSED CRAWLER FOR DARK WEB FORUMS

2.1 Introduction

Extremist groups frequently use the web to promote hatred and violence (Glaser et al., 2002). This problematic facet of the Internet is often referred to as the Dark Web (Chen, 2006). An important component of the Dark Web is extremist forums hidden deep within the Internet. Many have stated the need for collection and analysis of Dark Web forums (Burris et al., 2000; Schafer, 2002). Dark Web materials have important implications for intelligence and security informatics related application (Chen, 2006). The collection of such content is also important for studying and understanding the diverse social and political views present in these online communities.

Crawlers are programs that can create a local collection or index of large volumes of web pages (Cho and Garcia-Molina, 2000). Crawlers can be used for general purpose search engines or for domain specific collection building. The latter are referred to as focused or topic driven crawlers (Chakrabarti et al., 1999; Pant et al., 2002).

There is a need for a focused crawler that can collect Dark Web forums. Many previous focused crawlers have focused on collecting static English web pages from the “surface web.” A Dark Web forum focused crawler faces several design challenges. One major concern is accessibility. Web forums are dynamic and often require memberships. They are part of the Hidden Web (Florescu et al., 1998; Raghavan and Garcia-Molina, 2001) which is not easily accessible through normal web navigation or standard crawling. There are also multilingual web mining considerations. More than 30% of the web is in non-English languages (Chen and Chau, 2003). Consequently, the Dark Web also
encompasses numerous languages. Another important concern is content richness. Dark web forums contain rich content used for routine communication and propaganda dissemination (Abbasi and Chen, 2005a; Zhou, Reid, et al., 2005). These forums contain static and dynamic text files, archive files, and various forms of multimedia (e.g., images, audio, and video files). Collection of such diverse content types introduces many unique challenges not encountered with standard spidering of indexable (text based) files.

In this study we propose the development of a focused crawler that can collect Dark Web forums. Our spidering system uses breadth and depth first (BFS and DFS) traversal based on URL tokens, anchor text, and link levels, for crawl space URL ordering. We also utilize incremental crawling for collection updating using wrappers to identify updated content. The system also includes design elements intended to overcome the previously mentioned accessibility, multilingual, and content richness challenges. Our system also includes tailored spidering parameters and proxies for each forum in order to improve accessibility. The crawler uses language-independent features for crawl space URL ordering in order to negate any complications attributable to the presence of numerous languages. We also incorporate iterative collection of incomplete downloads and relevance feedback for improved multimedia collection.

The remainder of the chapter is organized as follows. Section 2.2 presents a review of related work on focused and hidden web crawling. Section 2.3 describes research gaps and our related research questions. Section 2.4 describes a research design geared towards addressing those questions. Section 2.5 presents a detailed description of our Dark Web forum spidering system. Section 2.6 describes experimental results evaluating
the efficacy of our human assisted approach for gaining access to Dark Web forums as well as the incremental update procedure that uses recall improvement. This section also highlights the Dark Web forum collection statistics for data gathered using the proposed system. Section 2.7 presents a case study conducted to illustrate the value of the collected dark web forums for content analysis while Section 2.8 contains concluding remarks.

2.2 Related Work: Focused and Hidden Web Crawlers

Focused crawlers “seek, acquire, index, and maintain pages on a specific set of topics that represent a narrow segment of the web” (Chakrabarti et al., 1999). The need to collect high quality domain-specific content results in several important characteristics for such crawlers that are also relevant to collection of Dark Web forums. Some of these characteristics are specific to focused and/or hidden web crawling while others are relevant to all types of spiders. We review previous research pertaining to these important considerations, which include accessibility, collection type and content richness, URL ordering features and techniques, and collection update procedures.

Before describing each of these issues in greater detail, we briefly discuss their importance for Dark Web forum crawling. Accessibility is an important consideration because Dark Web forums often require membership to access member postings (Chen, 2006). Furthermore, Dark Web forums are rich in multimedia content, including images and videos (Abbasi and Chen, 2005a; Zhou, Reid, et al., 2005). URL ordering features and techniques ensure that only the desired pages are collected, and in the most efficient manner (Guo et al., 2006). Different collection-update procedures have important implications for overall collection recall.
2.2.1 Accessibility

Most search engines cover what is referred to as the “publicly indexable Web” (Lawrence and Giles, 1998; Raghavan and Garcia-Molina, 2001). This is the part of the web easily accessible with traditional web crawlers (Sizov et al., 2003). As noted by Lawrence and Giles (1998), a large portion of the Internet is dynamically generated. Such content typically requires users to have prior authorization, fill out forms, or register (Raghavan and Garcia-Molina, 2001). This covert side of the Internet is commonly referred to as the hidden/deep/invisible web. Hidden web content is often stored in specialized databases (Lin and Chen, 2002). For example, the IMDB movie review database contains a plethora of useful information regarding movies; yet standard crawlers cannot access this information (Sizov et al., 2003). A study conducted in 2000 found that the invisible web contained 400-550 times the information present in the traditional surface web (Bergman, 2000; Lin and Chen, 2002).

Two general strategies have been introduced to access the hidden web via automated web crawlers. The first approach entails use of automated form filling techniques. Several different automated query generation approaches for querying such “hidden web” databases and fetching the dynamically generated content have been proposed (e.g., Barbosa and Freire, 2004; Ntoulas et al., 2005). Other techniques keep an index of hidden web search engines and redirect user queries to them (Lin and Chen, 2002) without actually indexing the hidden databases. However, many automated approaches ignore/exclude collection or querying of pages requiring login (e.g., Lage et al., 2002).
Thus, automated form filling techniques seem problematic for Dark Web forums where login is often required.

A second alternative for accessing the hidden web is a task-specific human assisted approach (Raghvan and Garcia-Molina, 2000). This approach provides a semi-automated framework that allows human experts to assist the crawler in gaining access to hidden content. The amount of human involvement is dependent on the complexity of the accessibility issues faced. For example, many simple forms asking for name, email address, etc. can be automated with standardized responses. Other more complex questions require greater expert involvement. Such an approach seems more suitable for the Dark Web, where the complexity of the access process can vary significantly.

2.2.2 Collection Type

Previous focused crawling research has been geared towards collecting web sites, blogs, and web forums. There has been considerable research on collection of standard web sites and pages relating to a particular topic, often for portal building. Srinivasan et al. (2002) and Chau and Chen (2003) fetched biomedical content from the web. Sizov et al. (2003) collected web pages pertaining to handicrafts and movies. Pant et al. (2002) evaluated their topic crawler on various keyword queries (e.g., “recreation”).

There has also been work on collecting weblogs. BlogPulse (Glance et al., 2004) is a blog analysis portal. The site contains analysis of key discussion topics/trends for roughly 100,000 spidered weblogs. Such blogs can also be useful for marketing intelligence (Glance et al., 2005a). Blogs containing product reviews analyzed using sentiment analysis techniques can provide insight into how people feel about various products.
Web forum crawling presents a unique set of difficulties. Discovering Web forums is challenging due to the lack of a centralized index (Glance et al., 2005a). Furthermore, Web forums require information-extraction wrappers for derivation of metadata (e.g., authors, messages, timestamps, etc.). Wrappers are important for data analysis and incremental crawling (i.e., re-spidering only those threads containing newly posted messages). Incremental crawling is discussed in greater detail in the “Collection-Update” section. There has been limited research on Web forum spidering. BoardPulse (Glance et al., 2005a) is a system for harvesting messages from online forums. It has two components: a crawler and a wrapper. Limanto et al. (2005) developed a Web forum information-extraction engine that includes a crawler, wrapper generator, and extractor (i.e., application of generated wrapper). Yih et al. (2004) created an online forum-mining system composed of a crawler and an information extractor for mining deal forums: forums where participants share information regarding deals or promotional events offered by online stores. The NetScan project (Smith, 2002) collected and visualized millions of pages from USENET newsgroups. RecipeCrawler (Li et al., 2006) is a focused crawler that collects cooking recipes from various information sources, including Web forums. RecipeCrawler uses the tree edit distance scores between Web pages to rank them in the crawl space (Li et al., 2006). Similar to BoardPulse (Glance et al., 2005a), RecipeCrawler also uses a crawler and a wrapper for extracting recipe information. Guo et al. (2006) proposed a board forum crawler that traverses board-based Web forums in a hierarchical manner analogous to that used by actual users manually browsing the forum. Their crawler uses Web page and URL token text features coupled with a rule-based
ranking mechanism to order URLs in the crawl space. Each of the aforementioned Web forum crawlers only collected pages from the Surface Web. There has been no prior research on collecting Dark Web forums, which requires the use of mechanisms for improving forum accessibility and collection recall.

2.2.3 Content Richness

The web is rich in indexable and multimedia files. Indexable files include static text files (e.g. HTML, Word and PDF documents) and dynamic text files (e.g., .asp, .jsp, .php). Multimedia files include images, animations, audio, and video files. Difficulties in indexing make multimedia content difficult to accurately collect (Baeza-Yates, 2003). Multimedia file sizes are typically significantly larger than indexable files, resulting in longer download times and frequent timeouts. Heydon and Najork (1999) fetched all MIME file types (including image, video, audio, and .exe files) using their Mercator crawler. They noted that collecting such files increased the overall spidering time and doubled the average file size as compared to just fetching HTML files. Consequently many previous studies have ignored multimedia content altogether (e.g., Pant et al., 2002).

2.2.4 URL Ordering Features

Aggarwal et al. (2001) pointed out four categories of features for crawl space URL ordering. These include links, URL and/or anchor text, page text, and page levels. Link based features have been used considerably in previous research. Many studies have used in/back links and out links (Cho et al., 1998; Pant et al., 2002). Sibling links (Aggarwal et al., 2001) consider sibling pages (ones with shared parent in link). Context graphs
(Diligenti et al., 2000) derive back links for each seed URL and use these to construct a multilayer context graph. Such graphs can be used to extract paths leading up to relevant nodes (target URLs). Focused/topical crawlers often use bag-of-words (BOW) found in the web page text (Aggarwal et al., 2001; Pant et al., 2002). For instance, Srinivasan et al. (2002) used BOW for biomedical text categorization in their focused crawler. While page text features are certainly very effective, they are also language dependent and can be harder to apply in situations where the collection is composed of pages in numerous languages. Other studies have also used URL/anchor text. Word tokens found within the URL anchor have been used effectively to help control the crawl space (Cho et al., 1998; Ester et al., 2001). URL tokens have also been incorporated in previous focused crawling research (Aggarwal et al., 2001; Ester et al., 2001). Another important category of features for URL ordering is page levels. Diligenti et al. (2000) trained text classifiers to categorize web pages at various levels away from the target. They used this information to build path models that allowed consideration of irrelevant pages as part of the path to attain target pages. A potential path model may consider pages one or two levels away from a target, known as tunneling (Ester et al., 2001). Ester et al. (2001) used the number of slashes “/” or levels from the domain as an indicator of URL importance. They argued that pages closer to the main page are likely to be of greater importance.

2.2.5 URL Ordering Techniques

Previous research has typically used breadth, depth, and best first search for URL ordering. Depth first (DFS) has been used in crawling systems such as Fish Search (De Bra and Post, 1994). Breadth first (BFS) (Cho et al., 1998; Ester et al., 2001; Najork and
Wiener, 2001) is one of the simplest strategies. It has worked fairly well in comparison with more sophisticated best-first search strategies (Cho et al., 1998; Najork and Wiener, 2001). However, BFS is typically not employed by focused crawlers that are concerned with identifying topic-specific web pages using the aforementioned URL ordering features.

Best-first uses some criterion for ranking URLs in the crawl space, such as link analysis or text analysis, or a combination of the two (Menczer, 2004). Numerous link analysis techniques have been used for URL ordering. Cho et al. (1998) evaluated the effectiveness of Page Rank and back link counts. Pant et al. (2002) also used Page Rank. Aggarwal et al. (2001) used the number of relevant siblings. They considered pages with a higher percentage of relevant siblings more likely to also be relevant. Sizov et al. (2003) used the HITS algorithm to compute authority scores while Chakrabarti et al. (1999) used a modified HITS. Chau and Chen (2003) used a Hopfield net crawler that collected pages related to the medical domain based on link weights.

Text analysis methods include similarity scoring approaches and machine learning algorithms. Aggarwal et al. (2001) used similarity equations with page content and URL tokens. Others have used the vector space model and cosine similarity measure (Pant et al., 2002; Srinivasan et al., 2002). Sizov et al. (2003) used support vector machines (SVM) with BOW for document classification. Srinivasan et al. (2002) used BOW and link structures with a neural net for ordering URLs based on the prevalence of biomedical content. Chen et al. (1998a; 1998b) used a genetic algorithm to order the URL crawl space for the collection of topic specific web pages based on bag-of-word representations.
of pages. Chakrabarti et al. (2002) incorporated an apprentice learner, which evaluated the utility of an outlink by comparing its HTML source code against prior training instances. Recent studies have compared the effectiveness of various machine learning classification algorithms such as Naïve Bayes, SVM, and Neural Networks, for focused crawling (Pant and Srinivasan, 2005, 2006).

2.2.6 Collection Update Procedure

Two approaches for collection updating are periodic and incremental crawling (Cho and Garcia-Molina, 2000). Periodic Web forum crawling entails eventually updating the collection by re-spidering all forum pages (e.g., Guo et al., 2006). This is commonly done because it is often easier than figuring out which Web forum pages to refresh, especially since the if-modified-since header does not provide information about which boards, subboards, and threads within a Web forum have been updated. Although periodic crawling is simpler from a crawler design/development perspective, it makes the collection process time consuming and inefficient. Alternatively, gathering multiple versions of a collection may improve overall recall. Incremental Web forum crawlers gather new and updated content by fetching only those boards and threads that have been updated since the forum was last collected (Glance et al., 2005a; Li et al., 2006; Yih et al., 2004). Incremental Web forum crawlers require the use of a wrapper that can parse out the “last updated” dates for boards and threads (Yih et al., 2004). This information is typically contained in the page’s body text.
2.2.7 Summary of Previous Research

Table 2.1 provides a summary of selected previous research on focused crawling. The majority of studies have focused on collection of indexable files from the surface web. There have only been a few studies that performed focused crawling on the hidden web. Similarly, only a few studies have collected content from web forums. Most previous research on focused crawling has used bag-of-word (BOW), link, or URL token features coupled with a best-first search strategy for crawl space URL ordering. Furthermore, most prior research also ignored the multilingual dimension, only collecting content in a single language (usually English). Collection of Dark Web forums entails retrieving rich content (including indexable and multimedia files) from the hidden web in multiple languages. Dark Web forum crawling is therefore at the cross-section of several important areas of crawling research, many of which have received limited attention in prior research. The following section summarizes these important research gaps and provides a set of related research questions which are addressed in the remainder of the chapter.

Table 2.1: Selected Previous Research on Focused Crawling

<table>
<thead>
<tr>
<th>System Name and Study</th>
<th>Access</th>
<th>Collection Type</th>
<th>Content Richness</th>
<th>URL Ordering Features</th>
<th>URL Ordering Techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td>GA Spider (Chen et al., 1998a; 1998b)</td>
<td>Surface Web</td>
<td>Topic Specific Web Pages</td>
<td>Indexable Files Only</td>
<td>BOW</td>
<td>Best-First: Genetic Algorithm</td>
</tr>
<tr>
<td>Focused Crawler (Chakrabarti et al., 1999)</td>
<td>Surface Web</td>
<td>Topic Specific Web Pages</td>
<td>Indexable Files Only</td>
<td>BOW and Links</td>
<td>Hypertext Classifier and Modified HITS algorithm</td>
</tr>
<tr>
<td>Context Focused (Diligenti et al., 2000)</td>
<td>Surface Web</td>
<td>Topic Specific Web Pages</td>
<td>Indexable Files Only</td>
<td>BOW and Context Graphs</td>
<td>Best-First: Vector Space, Naïve Bayes, and Path Models</td>
</tr>
<tr>
<td>Intelligent Crawler (Aggarwal et al., 2001)</td>
<td>Surface Web</td>
<td>Topic Specific Web Pages</td>
<td>Indexable Files Only</td>
<td>BOW, URL Tokens, Anchor Text, Links</td>
<td>Best-First: Similarity Scores and Link Analysis</td>
</tr>
<tr>
<td>Name</td>
<td>Web Type</td>
<td>Web Pages Type</td>
<td>Indexable Files Only</td>
<td>Scoring Method/Technique</td>
<td></td>
</tr>
<tr>
<td>---------------------------</td>
<td>----------------</td>
<td>-------------------------------------</td>
<td>----------------------</td>
<td>------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Ariadne (Ester et al., 2001)</td>
<td>Surface Web</td>
<td>Topic Specific Web Pages</td>
<td>Indexable Files Only</td>
<td>BOW, URL Tokens, Anchor text, Links, User Feedback, Levels</td>
<td></td>
</tr>
<tr>
<td>Hidden Web Exposer (Raghavan and Garcia-Molina, 2001)</td>
<td>Hidden Web</td>
<td>Dynamic Search Forms</td>
<td>Indexable Files Only</td>
<td>URL Tokens</td>
<td></td>
</tr>
<tr>
<td>InfoSpiders (Srinivasan et al., 2002)</td>
<td>Surface Web</td>
<td>Biomedical Pages and Documents</td>
<td>Indexable Files Only</td>
<td>BOW and Links</td>
<td></td>
</tr>
<tr>
<td>NetScan (Smith, 2002)</td>
<td>Surface Web</td>
<td>USENET Web Forums</td>
<td>n/a</td>
<td>n/a</td>
<td></td>
</tr>
<tr>
<td>Topic Crawler (Pant et al., 2002)</td>
<td>Surface Web</td>
<td>Topic Specific Web Pages</td>
<td>Indexable Files Only</td>
<td>BOW</td>
<td></td>
</tr>
<tr>
<td>Hopfield Net Crawler (Chau and Chen, 2003)</td>
<td>Surface Web</td>
<td>Medical Domain Web Pages</td>
<td>Indexable Files Only</td>
<td>Links</td>
<td></td>
</tr>
<tr>
<td>BINGO! (Sizov et al., 2003)</td>
<td>Surface and Hidden Web</td>
<td>Handicraft and Movie Web Pages</td>
<td>Indexable Files Only</td>
<td>BOW and Links</td>
<td></td>
</tr>
<tr>
<td>BlogPulse (Glance et al., 2004)</td>
<td>Surface Web</td>
<td>Weblogs for various topics.</td>
<td>Indexable Files Only</td>
<td>Weblog Text</td>
<td></td>
</tr>
<tr>
<td>Hot Deal Crawler (Yih et al., 2004)</td>
<td>Surface Web</td>
<td>Online Deal Forums</td>
<td>Indexable Files Only</td>
<td>URL Tokens, Thread Date</td>
<td></td>
</tr>
<tr>
<td>BoardPulse (Glance et al., 2005)</td>
<td>Surface Web</td>
<td>Product Web Forums</td>
<td>Indexable Files Only</td>
<td>URL Tokens, Thread Date</td>
<td></td>
</tr>
<tr>
<td>Web Forum Spider (Limanto et al., 2005)</td>
<td>Surface Web</td>
<td>Web Forums</td>
<td>Indexable Files Only</td>
<td>Web Page Text and URL Tokens</td>
<td></td>
</tr>
<tr>
<td>Board Forum Crawler (Guo et al., 2006)</td>
<td>Surface Web</td>
<td>Board Web Forums</td>
<td>Indexable Files Only</td>
<td>Web Page Text and URL Tokens</td>
<td></td>
</tr>
<tr>
<td>RecipeCrawler (Li et al., 2006)</td>
<td>Surface Web</td>
<td>Recipe Sites, Blogs, and Web Forums</td>
<td>Indexable Files Only</td>
<td>Web Page Text</td>
<td></td>
</tr>
<tr>
<td>RecipeCrawler (Li et al., 2006)</td>
<td>Surface Web</td>
<td>Recipe Sites, Blogs, and Web Forums</td>
<td>Indexable Files Only</td>
<td>Web Page Text</td>
<td></td>
</tr>
<tr>
<td>Board Forum Crawler (Guo et al., 2006)</td>
<td>Surface Web</td>
<td>Board Web Forums</td>
<td>Indexable Files Only</td>
<td>Rule Based: Uses URL tokens and text</td>
<td></td>
</tr>
<tr>
<td>Hot Deal Crawler (Yih et al., 2004)</td>
<td>Surface Web</td>
<td>Online Deal Forums</td>
<td>Indexable Files Only</td>
<td>Date Comparison</td>
<td></td>
</tr>
<tr>
<td>BoardPulse (Glance et al., 2005)</td>
<td>Surface Web</td>
<td>Product Web Forums</td>
<td>Indexable Files Only</td>
<td>Wrapper learning of site structure</td>
<td></td>
</tr>
<tr>
<td>Web Forum Spider (Limanto et al., 2005)</td>
<td>Surface Web</td>
<td>Web Forums</td>
<td>Indexable Files Only</td>
<td>Machine Learning Classifier</td>
<td></td>
</tr>
<tr>
<td>RecipeCrawler (Li et al., 2006)</td>
<td>Surface Web</td>
<td>Recipe Sites, Blogs, and Web Forums</td>
<td>Indexable Files Only</td>
<td>Best-First: Tree Edit Distance Similarity Scores</td>
<td></td>
</tr>
</tbody>
</table>
2.3 Research Gaps and Questions

Based on our review of previous literature we have identified several important research gaps.

2.3.1 Focused Crawling of the Hidden Web

There has been limited focused crawling work on the hidden web. Most focused crawler studies developed crawlers for the surface web (Raghavan and Garcia-Molina, 2001). Prior hidden web research mostly focused on automated form filling or query redirection to hidden databases, i.e., accessibility issues. There has been little emphasis on building topic-specific web page collections from these hidden sources. We are not aware of any attempts to automatically collect Dark Web content pertaining to hate and extremist groups.

2.3.2 Content Richness

Most previous research has focused on indexable (text based) files. Large multimedia files (e.g., videos) can be hundreds of MB. This can cause connection timeouts or excessive server loads, resulting in partial/incomplete downloads. Furthermore, the challenges in indexing multimedia files pose problems. It’s difficult to assess the quality of collected multimedia items. As Baeza-Yates (2003) noted, automated multimedia indexing is more of an image retrieval challenge than an information retrieval problem. Nevertheless, given the content richness of the Internet in general and the Dark Web in specific (Chen, 2006), there is a need to capture multimedia files.
2.3.3 Collection Recall Improvement

Prior crawling research has not addressed the issues associated with collecting content in adversarial settings. Dark Web forum spidering involves avoiding detection since it could have obvious ramifications for collection recall.

2.3.4 Web Forum Collection-Update Strategies

There has been considerable research on evaluating various collection-update strategies for Web sites (e.g., Cho and Garcia-Molina, 2000); however, there has been little work done on comparing the effectiveness of periodic versus incremental crawling for Web forums. Most Web forum research has assumed an incremental approach. Given the accessibility concerns associated with Dark Web forums, periodic and incremental approaches both provide varying benefits. Periodic crawlers can improve collection recall by allowing multiple attempts at capturing previously uncollected pages. This may be less of a concern for Surface Web forums, but is important for the Dark Web. In contrast, incremental crawlers can improve collection efficiency and reduce redundancy. There is a need to evaluate the effectiveness of periodic and incremental crawling applied to Dark Web forums.

2.3.5 Research Questions

Based on the gaps described, we propose the following research questions:

1) How effectively can Dark Web forums be identified and accessed for collection purposes?

2) How effectively can Dark Web content (indexable and multimedia) be collected?
3) Which collection update procedure (periodic or incremental) is more suitable for Dark Web forums? How can recall improvement further enhance the update process?

4) How can analysis of extracted information from Dark Web forums improve our understanding of these online communities?

2.4 Research Design

2.4.1 Proposed Dark Web Forum Crawling System

In this study we propose a Dark web forum spidering system. Our proposed system consists of an accessibility component that uses a human-assisted registration approach to gain access to Dark Web forums. Our system also utilizes multiple dynamic proxies and forum specific spidering parameter settings to maintain forum access.

Our URL Ordering component uses language independent URL ordering features to allow spidering of Dark Web forums across languages. We plan to focus on groups from three regions: U.S. Domestic, Middle East, and Latin America/Spain. Additionally a rule based URL ordering technique coupled with BFS and DFS crawl space traversal is utilized. Such a technique is employed in order to minimize the amount of irrelevant web pages collected.

We also propose the use of an incremental crawler that uses forum wrappers to determine the subset of threads that need to be collected. Our system will include a recall improvement procedure that parses the spidering log and reinserts incomplete downloads into the crawl space. Finally, the system features a collection analyzer that checks
multimedia files for duplicate downloads and generates collection statistics at the forum, region, and overall collection levels.

2.4.2 Accessibility

As noted by Raghavan and Garcia-Molina (2001), the most important evaluation criterion for Hidden Web crawling is how effectively the content was accessed. They developed an accessibility metric as follows: databases accessed / total attempted. We intend to evaluate the effectiveness of the task-specific human assisted approach in comparison with not using such a mechanism. Specifically we would also like to evaluate our system’s ability to access Dark Web forums. This translates into measuring the percentage of attempted forums accessed.

2.4.3 Recall-Improvement Mechanism

Given the collection challenges regarding Dark Web forums, we propose the use of a recall-improvement mechanism that controls various spidering settings for enhanced collection recall. The recall-improvement component is intended to control key spidering parameters such as the number of spiders per forum, the proxies per spider, and other pertinent spidering settings. It is essentially a heuristic used to counterattack crawler detection.

2.4.4 Incremental Crawling for Collection Updating

We plan to evaluate the effectiveness of our proposed incremental crawler in comparison with periodic crawling. The incremental crawler will obviously be more efficient in terms of spidering time and data redundancy. However, a periodic crawling approach gets multiple attempts to collect each page, which can improve overall
collection recall. Evaluation of both approaches is intended to provide additional insight into which collection update technique is more suitable for Dark Web forum spidering.

2.5 System Design

Based on our research design, we implemented a focused crawler for Dark Web forums.

Our system consists of four major components (shown in Figure 2.1):

- Forum Identification: to identify the list of extremist forums to spider;
- Forum preprocessing: which includes accessibility and crawl space traversal issues as well as forum wrapper generation;
- Forum spidering: which consists of an incremental crawler and recall improvement mechanism;
- Forum storage and analysis: to store and analyze the forum collection.
2.5.1 Forum Identification

The forum identification phase has three components.

Step 1: Identify extremist groups

Sources for the US domestic extremist groups include the Anti-Defamation League (ADL), FBI, Southern Poverty Law Center (SPLC), Militia Watchdog (MW), and the Google Web Directory (GD) (as a supplement). Sources for the international extremist groups include the United States Committee for a Free Lebanon (USCFAFL), Counter-Terrorism Committee (CTC) of the UN Security Council (UN), US State Department
Due to regional and language constraints, we chose to focus on groups from three areas: North America (English), Latin-America (Spanish), and the Middle East. These groups are all significant for their socio-political important. Furthermore, collection and analysis of Dark Web content from these three regions can facilitate a better understanding of the relative social and cultural differences between these groups. In addition to obvious linguistic difference, groups from these regions also display different web design tendencies and usage behavior (Abbasi and Chen, 2005a) which provide a unique set of collection and analysis challenges.

Step 2: Identify forums from extremist websites

We identify an initial set of extremist group URLs, and then use link analysis for expansion purposes as shown in Figure 2.2. The initial set of URLs is identified from three sources: Firstly we use search engines coupled with a lexicon containing extremist organization name(s), leader(s)’ and key members’ names, slogans, and special keywords used by extremists. Secondly we utilize government reports. Finally, we reference research centers. A link analysis approach is used to expand the initial list of URLs. We incorporate a backlink search using Google, which has been shown to be effective in prior research (Diligenti et al., 2000). Outlinks for initial seed URLs as well as their inlinks identified using Google also are collected. The identified Web forums are manually checked by domain experts. Only verified Dark Web forums are collected.
Step 3: Identify forums hosted on major web sites

We also identify forums hosted by other web sites and public internet service providers (ISPs) that are likely to be used by Dark Web groups. For example MSN groups, AOL Groups, etc. Public ISPs are searched with our Dark Web domain lexicon for a list of potential forums.

The above three steps help identify a seed set of Dark Web forums. Once the forums have been identified, several important preprocessing issues must be resolved before spidering. These include accessibility concerns and identification of forum structure in order to develop proper features and techniques for managing the crawl space.

2.5.2 Forum Preprocessing

The forum preprocessing phase has three components: accessibility, structure, and wrapper generation. The accessibility component deals with acquiring and maintaining
access to Dark Web forums. The structure component is designed to identify the forum URL mapping and devise the crawl space URL ordering using the relevant features and techniques.

2.5.2.1 Forum Accessibility

Step 1: Apply for membership

Many Dark Web forums do not allow anonymous access (Zhou, Reid, et al., 2005). In order to access and collect information from those forums one must create a user ID and password, send an application request to the web master, and wait to get permission/registration to access the forum. In certain forums, web masters are very selective. It can take a couple of rounds of emails to get access privilege. For such forums, human expertise is invaluable. Nevertheless, in some cases, access cannot be attained.

Step 2: Identify appropriate spidering parameters

Spidering parameters such as number of connections, download intervals, timeout, speed, etc., need to be set appropriately according to server and network limitations and the various forum blocking mechanisms. Dark Web forums are rich in terms of their content. Multimedia files are often fairly large in volume (particularly compared to indexable files). The spidering parameters should be able to handle download of larger files from slow servers. However we may still be blocked based on our IP address. Therefore, we use proxies to increase not only our recall but also our anonymity.

Step 3: Identify appropriate proxies
We use three types of proxy servers, as shown in Figure 2.3. Transparent proxy servers are those that provide anyone with your real IP address. Translucent proxy servers hide your IP address or modify it in some way to prevent the target server from knowing about it. However, they let anyone know that you are surfing through a proxy server. Opaque proxy servers (preferred) hide your IP address and do not even let anyone know that you are surfing through a proxy server. There are several criteria for proxy server selection, including the latency (the smaller the better), reliability (the higher the better), and bandwidth (the faster the better). We update our list of proxy servers periodically from various sources, including free proxy providers such as www.xroxy.com and www.proxy4free.com. Additionally, the crawler uses a Web browser user agent string and does not follow the robot exclusion protocol, though nearly none of the Dark Web forums collected had a robots.txt file.

![Diagram of proxy servers](image)

Figure 2.3: Proxies Used for Dark Web Forum Crawling

2.5.2.2 Forum Structure

Step 1: Identify site maps
We first identify the site map of the forum based on the forum software packages. Glance et al. (2005a) noted that although there are only a handful of commonly used forum software packages, they are highly customizable. Forums typically have hierarchical structures with boards, threads, and messages (Glance et al., 2005a; Yih et al., 2004). They also contain considerable additional information such as message-posting interfaces, search, printing, advertisement, and calendar pages (all irrelevant from our perspective). Furthermore, forums contain multiple views of member postings (e.g., sorted by author, date, topic, etc.). Collecting these duplicate views can introduce considerable redundancy into the collection, dramatically increase collection time, increase the likelihood of being detected/blocked, and result in spider traps (Guo et al., 2006). The URL ordering features and techniques are important to allow the crawler to collect only the desired pages (i.e., ones containing non-redundant message postings) in the most efficient manner.

Step 2: URL Ordering Features

Our spidering system uses two types of language independent URL ordering features, URL tokens and page levels. With respect to URL tokens, for web forums, we’re interested in URLs containing words such as “board,” “thread,” “message” etc. (Glance et al., 2005a). Additional relevant URL tokens include domain names of third party file hosting web sites. These third parties often contain multimedia files. File extension tokens (e.g., “.jpg” and “.wmv”) are also important. URLs that contain phrases such as “sort=voteavg” and “goto=next” are also found in relevant pages. However these are not unique to board, thread, and message pages, hence such tokens are not considered
significant. The set of relevant URL tokens differs based on the forum software being used. Such tokens are language independent yet software specific.

Page levels are also important as evidenced by prior focused crawling research (Diligenti et al., 2000; Ester et al., 2001). URL level features are important for Dark Web forums due to the need to collect multimedia content. Multimedia files are often stored on third party host sites that may be a few levels away from the source URL. In order to capture such content, we need to use a rule based approach that allows the crawler to go a few additional levels. For example, if the URL or anchor text contains a token that is a multimedia file extension or the domain name for a common third party file carrier, we want to allow the crawler to “tunnel” a few links.

Step 3: URL Ordering Techniques

As mentioned in the previous section, we use rules based on URL tokens and levels to control the crawl space. Moreover to adapt to different forum structures, we need to use different crawl space traversal strategies. Breadth first (BFS) is used for board page forums while depth first (DFS) for internet service provider (ISP) forums. DFS is necessary for many ISP forums due to the presence of ad pages that periodically appear within these forums. When such an ad page appears it must be traversed in order to get to the message pages (typically the ad pages have a link to the actual message page). Figure 2.4 illustrates how the BFS and DFS are performed for each forum type. Only the colored pages are fetched while the number indicates the order in which the pages are traversed by the crawler. One level of tunneling is allowed to fetch multimedia content hosted on third-party host Web sites outside of the Web forum. A parser analyzes the URL tokens
and anchor text for multimedia keywords. These include (a) the domain names for popular third-party hosts (b) multimedia file extensions such as .wmv, and .avi; (c) terms appearing in the anchor text, such as “video,” “movie,” and “clip.” Only URLs containing attributes from the aforementioned feature categories are tunneled.

Figure 2.4: URL Traversal Strategies

2.5.2.3 Wrapper Generation

Forums are dynamic archives that keep historical messages. It is beneficial to only spider newly posted content when updating the collection. This is achieved by generating wrappers that can parse web forum board and thread pages (Glance et al., 2005b). Board pages tell us when each thread was last updated with new messages. Using this information, one may respider only those thread pages containing new postings (Guo et al., 2006). Web forums generally use a dozen or so popular software for creating Web forums, including vBulletin, Crosstar, DCForum, ezBoard, Invision, phpBB, and so on. We developed wrappers based on these forums’ templates, as was done by previous research (e.g., Glance et al., 2005a; Guo et al., 2006). The wrappers parse out the board
pages and compare the posting dates for the most recent messages for all threads in a forum against the dates when the threads were last collected. If the thread has been updated, an incremental crawler retrieves all new pages (i.e., it fetches all pages containing messages posted since the thread was last spidered). The use of an incremental crawler via wrappers is an efficient way to collect Web forum content (Guo et al., 2006).

2.5.3 Forum Spidering

Figure 2.5 below shows the spidering process. The incremental crawler fetches only new and updated threads and messages. A log file is sent to the recall improvement component. The log shows the spidering status of each URL. A parser is used to determine the overall status for each URL (e.g., “download complete,” “connection timed out”). The parsed log is sent to the log analyzer which evaluates all files that weren’t downloaded. It determines whether the URLs should be respiered.

![Spidering Process Diagram](image)

Figure 2.5: Spidering Process

Figure 2.6 shows sample entries from the original and parsed log. The original log file shows the download status for each file (URL). The parsed log shows the overall status as
well as the reason for download failure (in the case of undownloaded files). Blue colored entries relate to downloaded files while red colored entries relate to undownloaded files. The log analyzer determines the appropriate course of action based on this cause of failure. “File Not Found” URLs are removed (not added to respidering list) while “Connection Timed Out” URLs are respidered. The recall improvement phase also checks the file sizes of collected web pages for partial/incomplete downloads. Multimedia file downloads are occasionally manually downloaded, particularly larger video files that may otherwise timeout.

<table>
<thead>
<tr>
<th>Log</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Parsed Log</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection Timed out: <a href="http://news.stcom.net/file=viewtopic&amp;f=2121,">http://news.stcom.net/file=viewtopic&amp;f=2121,</a></td>
</tr>
</tbody>
</table>

Figure 2.6: Example Log and Parsed Log Entries

Once the list of re-spidering URLs has been generated, the recall-improvement mechanism adjusts important spidering settings to improve collection performance. There are several important spidering parameters that can have an impact on Dark Web forum collection recall. These include the number of spiders per forum and the total number of proxies and proxies per spider as well as the batch size (i.e., the subset of URLs to be collected at a time) and timeout interval between batches. Given the large number of
potential URLs that may need to be fetched from a single forum, URLs in the crawl space are broken up into batches to alleviate forum server overload. Spidering parameters are adjusted based on the premise that the uncollected pages (requiring re-spidering) likely failed to be retrieved due to excessive load on the forum server or as a result of being blocked by the network or forum administrator. Therefore, the recall-improvement mechanism decreases the number of spiders and URLs per batch while also increasing the number of proxies per spider and the timeout interval between batches. These spidering adjustments are made to alleviate server load and avoid blockages. The steps involved in the spidering adjustment component of the recall-improvement mechanism are shown next. The values in parentheses signify the possible range of values for that particular parameter. For instance, a new forum would initially be crawled using 60 spiders; however, if necessary, this number may eventually decrease to 1 to improve recall.

1. Decrease the number of spiders per forum by half (1–60).
2. Increase the proxy ratio (i.e., No. of proxies per spider) by 1 (1–5).
3. Decrease the number of URLs per batch by half (100–1000).
4. Increase the timeout interval between batches by 5 s (5–60).

2.5.4 Forum Storage and Analysis

The forum storage and analysis phase consists of a statistics generation and duplicate multimedia removal components.
2.5.4.1 Statistics Generation

Once files have been collected, they must be stored and analyzed. The statistics consist of four major categories:

- Indexable files: HTML, Word, PDF, Text, Excel, PowerPoint, XML, and
  Dynamic files (e.g., PHP, ASP, JSP).
- Archive files: RAR, ZIP.
- Non-standard files: Unrecognized file types.

2.5.4.2 Duplicate Multimedia Removal

Dark Web forums often share multimedia files, but the names of those files may be changed. Moreover, some multimedia files’ suffixes are changed to other file types’ suffixes, and vice versa. For example, an HTML file may be named as a “.jpg.” Therefore, simply relying on file names results in inaccurate multimedia file statistics. We use an open-source duplicate multimedia removal software tool that identifies multimedia files by their meta data encoded into the file, instead of their suffixes (file extensions). It compares files based on their MD5 (Message-Digest algorithm 5) values, which are the same for duplicate video files collected from various Internet sources. MD5 is a widely-used cryptographic hash function with a 128-bit hash value. Comparing MD5 values allows a more accurate mechanism for differentiating multimedia files than does simply comparing file names, types, and sizes. In our analysis of duplicate Dark Web multimedia files, comparing MD5 hashes found three times as many duplicates as simply relying on file names, sizes, and types.
2.5.5 Dark Web Forum Crawling System Interface

Figure 2.7 shows the interface for the proposed Dark Web Forum spidering system. The interface has four major components. The “Forums” panel in the top left shows the spidering queue in a table that also provides information such as the forum name, URL, region, when it was last spidered, and whether the forum is still active. The “Spidering Status” panel in the top right corner displays information about the percentage of board, sub-board, and thread pages collected for the current forum being spidered. The “Forum Statistics” panel in the bottom left shows the quantity and size of the various file type collected for each forum, using tables, pie charts, and parallel coordinates. The “Forum Profile” panel in the bottom left shows each forum’s membership information and forum spidering parameters, including the number of crawlers, URL ordering technique (i.e., BFS or DFS), and URL ordering features (e.g., URL tokens, keywords) used to control the crawl space.
2.6 Evaluation

We conducted three experiments to evaluate our system. The first experiment involved assessing the effectiveness of our human-assisted accessibility mechanism. Raghavan and Garcia-Molina (2001) noted that accessibility is the most important evaluation criterion for Hidden Web research. We describe how effectively we were able to access Dark Web forums in our collection efforts using the human-assisted approach in comparison with standard spidering without any accessibility mechanism.

The second experiment assessed the impact of different spidering parameter settings on collection recall. Since accessibility and recall of Dark Web forum content is a critical
concern, we evaluated the impact on collection recall of using a different number of spiders per forum, proxies per spider, batch sizes, and timeout intervals between batches.

The third experiment entailed evaluating the proposed incremental spidering approach that uses recall improvement as a collection-updating procedure. We performed an evaluation of the effectiveness of periodic crawling as compared to standard incremental crawling and our incremental crawler, which uses iterative recall improvement for Dark Web forum collection updating.

For the latter two experiments, we used precision, recall, and F-measure to evaluate performance. For Web forums, relevant documents were considered to be unique Web pages containing forum postings (Glance et al., 2005a). Since Web forums are dynamic, their postings can be arranged in numerous ways (e.g., by date, by topic, by author, etc.). From a collection perspective, these views contain duplicate information: Only a single copy of each posting is desired (Guo et al., 2006). Irrelevant pages include ones containing duplicate forum postings or no forum postings at all as well as incorrectly collected pages (i.e., ones containing an HTML error code). Hence, consistent with prior forum crawling research (Glance et al., 2005a), we define precision, recall, and F-measure as follows:

\[ a = \text{No. of retrieved pages containing nonduplicate forum postings} \]
\[ b = \text{Total no. of pages containing nonduplicate forum postings} \]
\[ c = \text{No. of retrieved pages containing duplicate forum postings} \]
\[ d = \text{No. of retrieved pages containing an HTML error code} \]
\[ e = \text{No. of retrieved pages that do not contain forum postings} \]
Recall = \frac{a}{b}

Precision = \frac{a}{(a + c + d + e)}

F-measure = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}

2.6.1 Forum Accessibility Experiment

Table 2.2 presents results on our ability to access Dark Web forums with and without a human-assisted accessibility mechanism. Using the human-assisted accessibility approach, we were able to access over 82% of Dark Web forums hosted by various ISPs and virtually all of the attempted stand-alone forums. The overall results (>91% accessibility) indicate that the use of a human-assisted accessibility mechanism provided good results for Dark Web forums. In contrast, using standard spidering without any accessibility mechanism resulted in only 59.66% of the forums being accessible to collect. The largest impact of the accessibility approach occurred on the hosted forums, where lack of usage of human-assisted accessibility resulted in a 34% drop in the number of forums collected (n=18).

Table 2.2: Dark Web Forum Accessibility Statistics

<table>
<thead>
<tr>
<th></th>
<th>Human Assisted Accessibility</th>
<th>Standard Spidering</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Hosted Forums</td>
<td>Stand Alone Forums</td>
</tr>
<tr>
<td>Total Attempted</td>
<td>52</td>
<td>67</td>
</tr>
<tr>
<td>Accessed/Collected</td>
<td>43</td>
<td>66</td>
</tr>
<tr>
<td>Inaccessible</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>% Collected</td>
<td>82.69%</td>
<td>98.51%</td>
</tr>
</tbody>
</table>

Pairwise t tests were conducted to assess the improved access performance of the human-assisted accessibility mechanism as compared to a standard spidering scheme.
devoid of any special accessibility method. The improved performance was statistically
significant ($\alpha=0.01$) for total performance as well as for both forum types ($p_s<0.001$).

2.6.2 Spidering Parameter Experiment

To evaluate the effectiveness of different settings for key spidering parameters, we
conducted a simulated experiment in which 40 Dark Web forums were spidered several
times using different parameter settings. The parameters of interest included the number
of spiders per forum, the number of proxies per spider, the number of URLs per batch,
and the timeout interval between batches. The less aggressive settings were run earlier
(e.g., using fewer spiders, longer timeout intervals, etc.) to decrease the likelihood of
forum administrators blocking the latter spidering runs. Figure 2.8 shows the average
percentage recall for different combinations of number of spiders and proxies per spider
applied to the 40 testbed forums. Each condition was run using a constant batch size (300
URLs) and timeout interval between batches (20 s). The figure can be read as follows:
When using 30 spiders per forum and one proxy per spider (i.e., 30 proxies in total), the
recall was slightly higher than 50%. In contrast, when using 30 spiders per forum and
five proxies per spider (i.e., 150 proxies total), the recall was slightly higher than 70%.

Based on the results in Figure 2.8, note that the use of proxies has a profound impact
on collection performance. Unlike regular forums, collection of Dark Web forums has
recall of less than 30% when no proxies are used because of aggressive blocking from the
forum masters. Recall constantly improves as the number of proxies per spider is
increased up to four, but levels off after that point with no significant improvement when
using five proxies per spider. This suggests that the use of four times as many proxies as spiders per forum provides a sufficient level of anonymity.

![Figure 2.8: Recall Results for Different Settings of Number of Spiders and Proxies per Spider](image)

The number of spiders per forum also impacts recall, with optimum recall attained using 20 spiders per forum. Using fewer spiders diminishes recall because of the extended duration required to spider the URLs in a batch, which causes the spiders to get detected. The use of more than 20 spiders (e.g., 30) decreases performance due to an excessive number of connections that can either alert the forum master and/or network administration or cause the server to overload. Thus, when selecting the number of spiders per forum, one must balance the time required to collect the pages with the amount of server load at any point in time. Using too few or too many spiders can decrease recall due to the time taken or the excessive server load, respectively. This finding was supported by an analysis of the log files when using a different number of spiders per forum. Figure 2.9 shows the number of uncollected pages from our 40 Web forum testbed, for different numbers of spiders when using five proxies per spider.
Uncollected pages were placed into two categories based on their spider log entries. “Connection time out” pages are those that could not be collected because our spider was connected to the forum for too long. “Limit exceeded” pages are those that could not be collected because the forum blocked the spider for exceeding its download quota for a particular time period. Note that using a smaller number of spiders results in greater connection timeouts whereas the use of 30 spiders leads to increased “limit exceeded” errors. The use of 20 spiders provides the optimal balance between the two types of errors.

![Figure 2.9: Number of Uncollected Pages for Different Numbers of Spiders](image)

We also tested the impact of different batch sizes and timeout intervals (between batches) on collection recall, using the same 40 Dark Web forum testbed. For this experiment, a constant number of spiders per forum (n=20) and proxies per spider (n=4) were incorporated for each combination of batch and timeout interval. The results are presented in Figure 2.10. The diagram can be read as follows: When using a 10-s timeout interval between batches and a 200 URL batch size, recall of approximately 65% was attained.
Based on the results in Figure 2.10, note that both batch size and timeout interval impact recall for Dark Web forums. Not surprisingly, longer timeout intervals equate to enhanced recall; there is a 20% improvement in performance when using a timeout interval of 30 s between batches as opposed to 5-s timeout interval. Additionally, larger batch sizes also lead to deteriorating performance. When using a 30-s timeout, the drop in recall is most noticeable when increasing the batch size from 300 to 400 URLs. Although smaller batch sizes and longer timeout intervals improve recall, they also increase the spidering time. Thus, using a batch size of 300 URLs with a timeout interval of 30 s may be more favorable since it can drastically reduce spidering time with a minimal drop in recall, as compared to using a batch size of 100 or 200 URLs. The parameter-testing experiments have important implications for the spidering of Dark Web forums. Based on the results, it appears that tuning of various spidering parameters, including the number of spiders, number of proxies per spider, batch size, and timeout interval, play an integral role in recall performance.

Figure 2.10: Recall Results for Different Settings of Batch Size and Timeout Interval
2.6.3 Forum Collection-update Experiment

To evaluate the effectiveness of the proposed incremental crawling with recall improvement approach (referred to as incremental + RI) for collection updating, we conducted a simulated experiment in which 40 Dark Web forums were spidered three times over a 3-month period between December 2007 and February 2007. Figure 2.11 shows the number of cumulative Web pages and the amount of new pages appearing in the 40 testbed forums across the 3-month period. There were approximately 128,000 unique Web pages in the testbed, which were used as the gold standard for precision, recall, and F-measure computation. We collected the pages on a monthly basis (a total of three iterations) using periodic, incremental, and incremental + RI collection-update procedures. The periodic crawler collected all pages in each iteration (the cumulative amounts in Figure 2.11) while the incremental crawler only collected the new pages for each iteration (the iterative amounts in Figure 2.11). The advantage of periodic crawling is the ability to ascertain multiple versions of a page, which can improve the likelihood of gathering pages uncollected in the previous round at the expense of collection time and server congestion. The incremental + RI procedure also collected the new pages, but used a recall mechanism that allowed improperly retrieved pages to be refetched n number of times. The recall-improvement phase, which identifies uncollected pages based on their spidering status and file size, is intended to retrieve uncollected pages in an efficient manner (i.e., without putting excessive burden on the forum servers). Consequently, a value of n=2 was utilized since we have found that excessive attempts (i.e., larger values of n) typically decrease performance due to server congestion. For all experimental
conditions, we used 20 spiders per forum, four proxies per spider, a batch size of 300 URLs per forum, and a timeout interval of 30 s.

![Figure 2.11: Number of Web Pages in Test Bed across 3 Months/Iterations](image)

Performance was evaluated using the precision, recall, and F-measures. Precision was defined as the percentage of pages downloaded that were correctly collected. Correctly collected pages included all relevant pages completely downloaded. Incorrect pages were those that were partial/incomplete or irrelevant. Recall was defined as the percentage of relevant pages collected.

Table 2.3 shows the experimental results for the three collection procedures. The incremental + RI method achieved the highest precision, recall, and F-measure in a more efficient manner than did the periodic approach. The incremental update without recall improvement was the most efficient timewise; however, it only had an F-measure of roughly 55%. The results suggest that Dark Web forums require the use of a spidering strategy that entails multiple attempts to fetch uncollected pages.
Table 2.3: Macro-Level Results for Different Update Procedures

<table>
<thead>
<tr>
<th>Update Procedure</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Measure</th>
<th>Time (min.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Periodic</td>
<td>74.32</td>
<td>69.03</td>
<td>71.58</td>
<td>6,101</td>
</tr>
<tr>
<td>Incremental</td>
<td>57.80</td>
<td>53.69</td>
<td>55.67</td>
<td>4,855</td>
</tr>
<tr>
<td>Incremental + RI</td>
<td>79.59</td>
<td>74.74</td>
<td>77.09</td>
<td>5,758</td>
</tr>
</tbody>
</table>

Figure 2.12 shows the overall F-measure for the three collection-updating procedures after each spidering iteration. The diagram exemplifies the impact of making multiple attempts to collect unfetched pages. Note that the overall performance of periodic crawling improves dramatically during the second and third iterations since many of the previously uncollected Web pages are gathered. Since the incremental + IR method immediately retrieves such pages, it maintains a consistently higher level of performance, as compared to the other two methods.

Figure 2.12: Results by Iteration for Various Collection Update Procedures

2.6.4 Forum Collection Statistics

We used our spidering system for collection of Dark Web Forums in three regions. The spider was run incrementally for a 20 month period between 4/2005 and 12/2006.
The spider collected indexable, multimedia, archive (e.g., .zip, .rar), and non-standard files (e.g., those with unknown/unrecognized file extensions).

Table 2.4 below shows the number of forums collected per region. The collection consists of stand-alone and hosted forums. In general, the Middle Eastern groups tend to make greater use of stand-alone forums while the U.S. domestic forums are more evenly distributed between hosted and stand-alone forums.

Table 2.4: Dark Web Forum Collection Statistics

<table>
<thead>
<tr>
<th>Region</th>
<th>Hosted Forums</th>
<th>Stand Alone Forums</th>
<th>Total Forums</th>
</tr>
</thead>
<tbody>
<tr>
<td>Middle Eastern</td>
<td>21</td>
<td>50</td>
<td>71</td>
</tr>
<tr>
<td>Latin American</td>
<td>6</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>US Domestic</td>
<td>16</td>
<td>13</td>
<td>29</td>
</tr>
<tr>
<td>Total</td>
<td>43</td>
<td>66</td>
<td>109</td>
</tr>
</tbody>
</table>

Table 2.5 shows the detailed collection statistics categorized by file types. Our system was able to collect a rich assortment of indexable and multimedia files. It’s interesting to note the large quantities of dynamic and multimedia files. Static HTML files, which were predominant on the Internet ten years ago, have a minimal amount of usage in the Dark Web forums. Dynamic files outnumber static HTML files by a ratio of 10:1 while multimedia files (particularly images) are also present more often. This is partially attributable to the use of various forum software packages that generate dynamic thread pages (typically .php files).
Table 2.5: Dark Web Forum Collection File Statistics

<table>
<thead>
<tr>
<th></th>
<th>No. of Files</th>
<th>Volume (Bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indexable Files</td>
<td>3,001,194</td>
<td>140,878,063,124</td>
</tr>
<tr>
<td>HTML Files</td>
<td>283,578</td>
<td>2,942,658,681</td>
</tr>
<tr>
<td>Word Files</td>
<td>2,108</td>
<td>46,649,107</td>
</tr>
<tr>
<td>PDF Files</td>
<td>16</td>
<td>8,168,345</td>
</tr>
<tr>
<td>Dynamic Files</td>
<td>2,715,354</td>
<td>137,178,574,841</td>
</tr>
<tr>
<td>Text Files</td>
<td>657</td>
<td>2,249,471,937</td>
</tr>
<tr>
<td>Excel Files</td>
<td>1</td>
<td>177,152</td>
</tr>
<tr>
<td>PowerPoint Files</td>
<td>2</td>
<td>528,834</td>
</tr>
<tr>
<td>XML Files</td>
<td>26</td>
<td>466,706</td>
</tr>
<tr>
<td>Multimedia Files</td>
<td>423,749</td>
<td>25,833,258,770</td>
</tr>
<tr>
<td>Image Files</td>
<td>422,155</td>
<td>8,554,125,848</td>
</tr>
<tr>
<td>Audio Files</td>
<td>5,479</td>
<td>3,664,642,638</td>
</tr>
<tr>
<td>Video Files</td>
<td>6,115</td>
<td>13,614,490,284</td>
</tr>
<tr>
<td>Archive Files</td>
<td>801</td>
<td>621,721,139</td>
</tr>
<tr>
<td>Non-Standard Files</td>
<td>443,244</td>
<td>17,303,588,746</td>
</tr>
<tr>
<td>Total</td>
<td>3,868,988</td>
<td>185,017,574,960</td>
</tr>
</tbody>
</table>

2.7 Dark Web Forum Case Study

To provide insight into the utility of our collection for content analysis of Dark Web forums, we conducted a detailed case study. Such case studies, which have been used in prior related work (e.g., Glance et al., 2005b), are useful for illustrating the value of the collection as well as the Dark Web forum crawling system used to generate the collection. Our case study involved topical and interactional analysis of eight Dark Web forums from our collection. Topic and interaction analysis have been prevalent forms of content analysis in previous computer-mediated communication research. The dataset consisted of messages from eight domestic supremacist forums. Table 2.6 provides the number of authors and messages for each forum in the test bed, with a total of 650 authors and approximately ten thousand message postings.
Table 2.6: Domestic Supremacist Forum Test Bed

<table>
<thead>
<tr>
<th>Forum</th>
<th>Authors</th>
<th>Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Angelic Adolf</td>
<td>28</td>
<td>78</td>
</tr>
<tr>
<td>Aryan Nation</td>
<td>54</td>
<td>489</td>
</tr>
<tr>
<td>CCNU</td>
<td>2</td>
<td>429</td>
</tr>
<tr>
<td>Neo-Nazi</td>
<td>98</td>
<td>632</td>
</tr>
<tr>
<td>NSM World</td>
<td>289</td>
<td>7,543</td>
</tr>
<tr>
<td>Smash Nazi</td>
<td>10</td>
<td>66</td>
</tr>
<tr>
<td>White Knights</td>
<td>24</td>
<td>751</td>
</tr>
<tr>
<td>World Knights</td>
<td>35</td>
<td>223</td>
</tr>
<tr>
<td>Total</td>
<td>650</td>
<td>10,211</td>
</tr>
</tbody>
</table>

2.7.1 Topical Analysis

Evaluation of key topics of discussion can provide insight into the groups’ content as well as the inter-relations between the various forums. The vector-space model (tf x idf) was used to determine the word vectors for each author. The word vectors consisted of bag-of-words after stop/function words were removed. We then constructed a n x n matrix of similarity scores computed using the cosine measure across all 650 authors. The similarity matrix was visualized using a spring-embedding algorithm which belongs to the family of force directed placement algorithms. Such algorithms are common multidimensional scaling techniques in which the distance between objects is proportional to their similarity (with closer objects being more similar). Spring-embedding algorithms are a popular technique in information retrieval for viewing similarities between documents (Chalmers and Chitson, 1992; Leuski and Allan, 2000). Our implementation shows authors placed based on their cosine similarity scores. Author clusters were manually annotated with descriptions of major discussion (based on term co-occurrences).
Figure 2.13 shows the annotated author MDS projections based on discussion topic similarities. Each circle denotes an author while the circle color indicates the author’s forum affiliation. The gray transparent ovals indicate author clusters based on common discussion topics. Table 2.7 provides descriptions of each of these topic clusters.

Based on Figure 2.13 and Table 2.7, it appears that the NSM World, Neo-Nazi, and Angelic Adolf forums all have ties with the National Socialist Movement (NSM) party. Members of these groups are avidly discussing issues relating to the party. The NSM World forum is the largest in size (in terms of members and postings) but also has the most diversity in terms of topics. This forum is the leading news source, with the most
content relating to domestic and international stories and events relevant to its members.
Most of the smaller forums (e.g. White Knights, World Knights and Smash Nazi) are
predominantly conversational forums where members discuss/argue their opinions and
beliefs. Overall there is considerable topical overlap across forums indicating that the
authors of these various online communities are discussing similar matters.

Table 2.7: Description of Major Discussion Topics in Test Bed Forums

<table>
<thead>
<tr>
<th>Topic</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSM Party News</td>
<td>News about National Socialist Movement party meetings, rallies, anniversary celebrations, and internal party politics.</td>
</tr>
<tr>
<td>Hate Crime News</td>
<td>News about violent inter-racial domestic crimes involving white victims.</td>
</tr>
<tr>
<td>Politics and Religion</td>
<td>Discussion about religious beliefs, foreign and domestic policies, and political malcontent.</td>
</tr>
<tr>
<td>General Discussion and Opinions</td>
<td>Opinions and beliefs about different races and religions.</td>
</tr>
<tr>
<td>Aryan Books</td>
<td>Information about the availability of literature pertaining to Aryan beliefs (including books and newsletters).</td>
</tr>
<tr>
<td>Persian Content</td>
<td>Content written in Farsi. There is a considerable Persian following in the Nazi groups (though the vast majority contribute in English).</td>
</tr>
</tbody>
</table>

2.7.2 Interaction Analysis

Evaluation of participant interaction can provide insight into the interrelations between various forums. We constructed the author interaction network across the 8 test bed forums. The interaction network shows whom each individual’s messages are directed towards; as well as additional forum members that are referenced in the message text. Figure 2.14 shows the author interaction network for the 650 authors in our test bed. Each circle (network node) denotes an author while the circle color indicates the author’s forum affiliation. The lines (links) between author nodes indicate interaction between
those two authors. As mentioned above, interaction can be in the form of direct communication between the two authors (i.e., one replying to the other’s message) or via an indirect reference to the other author’s screen name. A spring-layout algorithm was used to cluster authors based on link/interaction strength.

Figure 2.14: Author Interaction Network for Domestic Supremacist Forums

The network provides evidence of considerable interaction between members across the various forums. Cross-forum interaction occurs when a message in one forum directly addresses a member of another forum. The only forums that do not have any such cross-forum interaction are CCNU and Smash Nazi. Coincidentally these are also the two
smallest forums in our test bed, with two and ten members respectively. In contrast, members of the NSM, Neo-Nazi, and Angelic Adolf forums have considerable interaction. This is consistent with the topical analysis presented in the previous section, which also found discussion topic similarities between members of these forums. These results are also consistent with previous Dark Web site analysis studies that found considerable linkage between various U.S. domestic supremacist web sites (Zhou, Reid, et al., 2005). The case study illustrates the utility of the Dark Web forum collection for content analysis of these online communities. Synchronous efforts to collect and analyze such web forum content are an important yet sparsely explored endeavor (Burris et al., 2000).

2.8 Conclusions and Future Directions

In this study we developed a focused crawler for collecting Dark Web forums. We used a human-assisted accessibility mechanism to access identified forums with a success rate of over 90%. Our crawler uses language independent features including URL tokens, anchor text, and level features, in order to allow effective collection of content in multiple languages. It also uses forum software specific traversal strategies and wrappers to support incremental crawling. The system uses an incremental crawling approach coupled with a recall improvement mechanism that continually re-spiders uncollected pages. Such an update approach outperformed the use of a standard incremental update strategy as well as the traditional periodic update method in a head-to-head comparison in terms of precision, recall, and computation time.
The system has been able to maintain up-to-date collections of 109 forums in multiple languages from three regions: U.S. domestic supremacist, Middle Eastern extremist and Latin groups. We also presented a case study using the collection in order to demonstrate its utility for content analysis. The case study provided insight into important discussion topics and interaction patterns for selected U.S. domestic supremacist forums. We believe that the proposed forum crawling system allows important entry to Dark Web forums which facilitates better accessibility for the analysis of these online communities. The collection of such content has significant academic and scientific value for the intelligence and security informatics as well as various other research communities interested in analyzing the social characteristics of Dark Web forums.

We have identified several important directions for future research. We plan to improve the Dark Web forum accessibility mechanism in order to attain higher access rates. We also plan to expand our collection efforts to also include weblogs and chatting log archives. Additionally, we intend to evaluate the effectiveness of multimedia categorization techniques to enhance our ability to collect relevant image and video content.
CHAPTER 3: SENTIMENTAL SPIDERING: LEVERAGING OPINION INFORMATION IN FOCUSED CRAWLERS

3.1 Introduction

The proliferation of user-generated content in Web 2.0 presents opportunities and challenges for decision making in various domains: many political, business intelligence (BI) and marketing intelligence (MI) applications could significantly benefit from (or in some cases, require) fast or even “real-time” analysis of relevant Web data. The development of effective and advanced focused crawlers remains critical due to the continual need for high-quality, relevant data collections that are manageable and efficient in terms of their creation, maintenance, update mechanism, and analyses (Pant and Srinivasan, 2009).

Previous work on focused crawling has primarily emphasized the collection of topic-relevant content and ignored the embedded opinion or sentiment information. However, a lot of Web 2.0 content is rich in opinion information and has obvious sentiment polarity (i.e., negative/positive/neutral sentiment) toward specific topics. It has stirred much excitement and created abundant opportunities for understanding the opinions toward social events, political movements, company strategies, marketing campaigns, and products (Chen and Zimbra, 2010). For instance, companies are increasingly interested in how they are perceived by environmental and animal rights groups in terms of corporate social responsibility (CSR) (Bhattacharya et al., 2009). Similarly, politicians are using the Web as a mechanism for gauging public sentiment (Wattal et al., 2010). Brand monitoring agencies have long sought ways to quickly “take the pulse” of consumers in regard to certain products. Knowledge of negative product sentiments can save firms
millions of dollars, and in some cases, can even save human lives (Subrahmanian, 2009). For instance, pharmaceutical companies are interested in learning about consumer accounts of adverse drug reactions in a timely manner due to the severe legal and monetary implications (Van Grootheest et al., 2003). Government and security agencies want to identify Web 2.0 users that are sympathetic to terrorism (Fu et al., 2010). Moreover, many individuals seek content published by people that share the same sentiment polarity on a topic, resulting in the phenomenon known as Cyber Balkanization (Van Alstyne and Brynjolfsson, 2005).

The increasing importance of sentiment information necessitates quick and efficient focused crawler methods to collect not only topic-relevant but also sentiment-relevant content from various Web 2.0 media such as the blogosphere, social network services (SNS), video-sharing sites, forums, etc. (Liu et al., 2010). Despite the prevalence of sentiment-related content on the Web (Wiebe, 1994), there has been limited work on focused crawlers capable of effectively collecting such content.

Actionable “real-time” intelligence requires balancing efficiency with accuracy. Focused crawlers incorporating information access refinements that improve precision without enhanced recall can be problematic. Decisions and judgements made using low-recall data collections can be heavily biased and lead to unexpectedly bad results. Focused crawlers that only evaluate the out-links of relevant pages and are likely to miss relevant pages that are not directly linked (Menczer et al., 2004). This problem is exacerbated when collecting content containing specific topics and sentiments due to the lower proportion of relevant pages to irrelevant ones (as compared to traditional topic
crawling tasks). Tunneling is a strategy utilized by focused crawlers to traverse irrelevant pages in order to reach relevant ones (Martin et al., 2001). In order to attain suitable recall levels, effective tunneling is essential for focused crawlers incorporating sentiment information. Suppose McDonald’s is interested in identifying and analysing negative opinions about their brand and/or products. In Figure 3.1, the first page describes design elements of McDonald’s logo. One of the out-links of this page provides a detailed description of McDonald’s logo history. This second page also provides a link to a page from www.mccruelty.com, a website with strong negative sentiments towards McDonald’s. This third page is obviously highly relevant to the collection task. However without tunneling, it would not be reached since a sentimental spider would only traverse the out-links of pages deemed relevant (and the first two pages are topically relevant but do not have relevant sentiment).

![Figure 3.1: Example Path for Tunneling](image)

The example presented in Figure 3.1 illustrates sentiment information and tunneling can improve focused crawlers’ abilities to collect opinionated content. Our goal in this chapter is to examine whether sentiment information is useful for crawling tasks that involve consideration of content encompassing opinions about a particular topic and to
explore how to use sentiment information for tunneling. We propose a novel focused crawler that incorporates topic and sentiment information as well as a graph-based tunneling mechanism for enhanced collection of opinion-rich web content regarding a particular topic. The crawler classifies web pages based on their topical and sentimental relevance and utilizes graph similarity information in tunneling. Experimental results demonstrate the effectiveness of our crawler over several comparison focused crawlers.

While Chapter 2 reviews and explores a wide range of focused crawler characteristics, in this chapter we mainly focus on two key characteristics, URL ordering features and URL ordering techniques. Chapter 2 addresses the data collection task in our CSI framework by contributing a solution for specific Web 2.0 medium, the Dark Web forum. In comparison, we innovatively expand existing URL ordering features with sentiment information, a feature that has been ignored by previous studies, and explore the sentiment spidering in this chapter.

The remainder of this chapter is organized as follows. Section 3.2 presents a brief review of existing work on focused crawling as well as research gaps. The proposed graph-based sentiment (GBS) crawler is discussed in Section 3.3. Section 3.4 describes the experimental test bed as well as the six comparison focused crawlers. This section also includes experimental results comparing GBS against the existing methods. Section 3.5 presents concluding remarks.

3.2 Literature Review

Focused crawlers aim to efficiently locate highly relevant target web pages by using available contextual information to guide the navigation of links and are seen as a way to
address the scalability limitations of universal search engines (Chakrabarti et al., 1999; Menczer et al., 2004). Two main characteristics of focused crawlers are the contextual information and the techniques used for candidate URL ordering and classification.

Three types of contextual information are useful for estimating the benefit of following a URL: link context, ancestor pages, and web graphs (Pant and Srinivasan, 2005). Link context refers to the lexical content around the URL in the page from which the URL was extracted (i.e., the parent page), which can range from text surrounding the link (called anchor text) to the whole content of the link’s parent page. Ancestor pages are the lexical content of pages that lead to the parent page of the URL. Web graphs refer to the hyperlink graphs comprised of in-links and out-links between web pages.

Link context is the most fundamental contextual information in classifier-based topical crawlers and has been utilized by most prior focused crawlers (Pant and Srinivasan, 2005). The popularity of the Vector Space Model (VSM) for text classification has also resulted in the use of VSM-based crawlers that rely exclusively on link context. They have been widely used in previous studies such as Aggarwal et al. (2001), Pant and Srinivasan (2005), and Menczer et al. (2004). A typical VSM-based crawler represents each web page as a vector space using the TF-IDF (term frequency and inverse document frequency) weighting schema (Salton and McGill, 1986). TF-IDF vector of a candidate web page is usually compared with vectors of relevant and irrelevant training pages in order to determine its relevance. Previous studies have also used a more selective keyword list as the basic vocabulary for the TF-IDF schema of VSM, which we refer to as Keyword-based crawler in this chapter (Menczer et al., 2004).
The quality of the keyword list is critical to the performance of a Keyword-based crawler. Domain experts may select keywords based on their domain knowledge. Conversely, automated feature selection techniques may be used to learn keywords that are adept at assessing the relevance of documents (Abbasi and Chen, 2008; Yang and Pedersen, 1997).

Crawlers that only rely on link context are often good at evaluating links of relevant pages, which is consistent with the topical locality hypothesis that claims similar content is more likely to be linked (Davison, 2000). However, the increased volume of Web data and the complex structure of the Web greatly reduces the recall of these crawlers because they fall short in learning tunneling strategies when relevant content is just a few links behind an apparently irrelevant page (Diligenti et al., 2000). Some researchers proposed to utilize external knowledge to broaden the search space if necessary, for example to temporarily change the crawling topic from “sailing” to “water sports” based on the hierarchical relationship between words (called “hyponymy”) (Martin et al., 2001). Such relationships can be identified using the Open Directory Project (ODP) or a lexical thesaurus such as WordNet (Martin et al., 2001).

Advanced crawling techniques have been developed to overcome the shortcomings of the above crawlers by utilizing the other two types of contextual information: ancestor page and web graph. Context Graph Model (CGM) is a good example of a crawler that incorporates ancestor pages in the crawling process (Diligenti et al., 2000). A context graph represents how a target document can be accessed from the web and consists of in-link pages and their ancestor pages. The CGM crawler builds Naïve Bayes classifiers for
each layer of the relevant training data’s context graph. These classifiers are then used to predict how far away an irrelevant page is from a relevant target page. Irrelevant pages are ranked in the queue based on their perceived proximity to relevant target pages. In head-to-head comparisons, the CGM crawler outperformed several focused crawlers that rely solely on link context information (Diligenti et al., 2000).

Among the three categories of contextual information exploited by focused crawlers, web graphs rely the least on the lexical content of a page. Pattern recognition refers to the act of determining to which category or class a given pattern belongs. Based on how patterns are represented, there are two types of pattern recognition: statistical and structural pattern recognition (Riesen and Bunke, 2010). In statistical pattern recognition, objects or patterns are represented by feature vectors. The abovementioned methods for evaluating link context and ancestor pages utilized statistical pattern recognition techniques. In contrast, structural pattern recognition utilizes symbolic data structures such as strings, trees, or graphs. Compared with feature vectors, graphs are better suited to describe spatial, temporal or conceptual relationships between objects. However, few search engines or focused crawlers have explored web graphs due to limitations in available graph information and computational constraints. Hopfield Net (HFN) (Chau and Chen, 2003; Chau and Chen, 2007) models the web graph as a weighted, single-layer neural network. It applies a spreading activation algorithm on the model to improve web retrieval. HFN outperformed breadth-first search (BFS) and PageRank (which also uses web graph information) in the collection of medical web pages. PageRank (Brin and Page, 1998), which is commonly used as a baseline in focused crawling studies, simulates
a random walk over the web taken by a web surfer and calculates the quality of a page proportionally to the quality of the pages that link to it. It attempts to identify hub nodes (i.e., pages that link many resourceful pages) in web graphs. Both HFN and PageRank use web graph information to pass accumulated weights to child pages (i.e., out-links).

Since classification techniques that using different types of contextual information have their own strengths and weaknesses, some researchers adopted ensemble techniques (Schapire and Singer, 1999; Allwein et al., 2001) and implemented various voting schemes that incorporated predictions from several classifiers. For example, Fürnkranz (2002) suggested four voting schemes: majority vote, weighted sum, weighted normalized sum, and maximum confidence. However, since ensemble techniques are time-consuming, they have mostly been used to evaluate crawler results. For instance, Pant and Srinivasan used a classifier ensemble comprised of eight Naïve Bayes, Support Vector Machines (SVM), and Neural Network classifiers to evaluate their crawlers (Pant and Srinivasan, 2005).

Based on our review of prior work on focused crawling, we have identified several research gaps. To the best of our knowledge, sentiment information has never been utilized by previous crawlers. Given the proliferation of user-generated content rich in opinions and sentiments, there remains a need to evaluate the efficacy of using sentiment information for enhanced focused crawling of opinion-rich web content regarding a particular topic. Moreover, several previous studies have pointed out that web graphs may provide essential cues about the merit of following a particular URL, resulting in improved tunneling capabilities (Broder et al., 2000; Pant and Srinivasan, 2005).
However, most studies have relied primarily on link context information to inform the navigation of the focused crawler. The few studies that used web graphs relied primarily on from-to linkage relations between parent-child nodes (Chau and Chen, 2003; Chau and Chen, 2007). Web graph structure has seen limited usage. In the following section, we describe the proposed graph-based sentiment crawler (GBS) that utilizes topic and sentiment information and graph-based tunneling to identify pages containing opinions about a particular topic.

3.3 Research Design

We propose a new focused crawler that can leverage sentiment information and labelled web graphs. Figure 3.2 illustrates its system design. This Graph-based sentiment crawler (GBS) consists of four modules: crawler, queue management, text classifier, and graph comparison. The first two modules are common to most focused crawlers. The queue management module ranks the current list of candidate URLs based on their weights. The weights associated with candidate URLs are determined by the last two modules (i.e., the text classifier and graph comparison), described in detail below. The crawler module crawls URLs in descending order based on their rank/location in the queue management module.
3.3.1 Text Classifier Module

Our text classifier module consists of a topic classifier and a sentiment classifier. Each classifier adopts a simple, computationally efficient categorization approach suitable for use within a focused crawler. The topic classifier computes the topical relevance of a page using a trained classification model, as follows. Given a set of training pages containing known relevant and irrelevant pages, we extract all non-stop word unigrams occurring at least 3 times. Each of these keywords \( a \) is weighted using the information gain heuristic, where a weight \( w(a) \) is computed based on the keyword’s level of entropy reduction (Shannon, 1948). Hence,
\[ w(a) = E(y) - E(y|a), \]
\[ E(y) = -\sum_{i \in y} p(y = i) \log_2 p(y = i) \]

where \( E(y) \) is the entropy across the set of classes \( y \) (i.e., relevant and irrelevant pages), and

\[ E(y | a) = -\sum_{j \in a} p(a = j) \sum_{i \in y} p(y = i | a = j) \log_2 p(y = i | a = j) \]

is the entropy of \( y \) given \( a \), where \( p(a=j) \) is the probability that keyword \( a \) has a value \( j \), where \( j \in \{0,1\} \) depending on whether or not \( a \) occurs in a particular web page. It is important to note that \( E(y) = 1 \) if the number of relevant and irrelevant pages in the training set are equal/balanced. For each keyword \( a \), we also compute its relevance \( r(a) \in \{-1, 1\} \), where \( r(a) = 1 \) if \( a \) occurs in a greater number of relevant training pages than irrelevant ones, \( r(a) = -1 \) otherwise. Once the topic classifier has been trained, it can be used to score a candidate page \( u \) as follows:

\[ TS(u) = \sum_a w(a)r(a)t(a) \]

where \( t(a) = 1 \) if keyword \( a \) occurs in page \( u \), \( t(a) = 0 \) otherwise. A candidate page \( u \) is considered topically relevant if \( TS(u) > 0 \).

The sentiment classifier computes a sentiment score, \( SS(u) \), for each candidate page \( u \). The sentiment classifier considers both sentiment polarities and intensities. Sentiment polarity pertains to whether a text has a positive, negative, or neutral semantic orientation (Abbasi et al., 2008). A given sentiment polarity (e.g., positive/negative) can also have varying intensities: for instance weak, mild, or strong. We utilize SentiWordNet (Esuli and Sebastiani, 2006), a lexical resource, to derive the sentiment polarities and intensities.
associated with the text surrounding relevant keywords contained in \( u \). SentiWordNet contains three sentiment polarity scores (i.e., positivity, negativity, objectivity) for synsets comprised of word-sense pairs (Esuli and Sebastiani, 2006). SentiWordNet contains scores for over 150,000 words, with scores being on a 0-1 scale. For instance, the synset consisting of the verb form of the word “short” and the word “short-change” has a positive score of 0 and a negative score of 0.75. As a preprocessing step, for each word \( w \) in SentiWordNet, we compute its semantic weight \( s(w) \) as the average of the sum of its positive and negative scores across word-sense pairs. To compute \( SS(u) \), we only consider the semantic weight of sentences containing relevant keywords found in \( u \). In other words, let \( B \) represent the subset of keywords found in \( u \) where \( r(b)=1 \) for each \( b \in B \). Further, let \( K_b \) denote the set of words from each sentence in \( u \) that contains \( b \). The sentiment score for each candidate page \( u \) is computed as the difference in semantic orientation between that page and the relevant pages in the training data set, regarding the words in \( B \). More specifically,

\[
SS(u) = \frac{1}{|B|} \sum_{b \in B} \left( \left( \frac{1}{|K_b|} \sum_{i \in K_b} s(i) \right) - \left( \frac{1}{|R_b|} \sum_{j \in R_b} s(j) \right) \right)
\]

where \( s(i) \) is the semantic score for word \( i \) and \( R_b \) denotes the set of words from each sentence in the relevant training pages that contains \( b \). Candidate page \( u \) is considered to contain relevant sentiment if \( SS(u) \) is less than a threshold parameter \( t \) (i.e., if \( SS(u) < t \)).

Figure 3.3 presents an illustration of the text classifier utilized by GBS. The top half of the figure shows the topic classifier, while the bottom half depicts the sentiment classifier. In the topic classifier, all keywords are indexed, weighted, and associated with
one of the two classes (based on their occurrence distribution across classes). In Figure 3.3, keywords associated with relevant pages are denoted by circles while ones associated with irrelevant pages are depicted by squares. Each candidate page is classified as relevant/irrelevant based on the sum of the weighted presence of these keywords. The sentiment classifier computes the difference in sentiment composition between candidate page sentences containing keywords associated with relevant pages (i.e., depicted by circles) and relevant training web page sentences containing those same keywords. Candidate pages that differ from the relevant pages by less than $t$ are considered to contain relevant sentiment information. By applying the text classifier module, each collected web page is categorized as belonging to one of the following four classes:

C1: Relevant topic and sentiment  C2: Relevant topic only

C3: Relevant sentiment only  C4: Irrelevant topic and sentiment

Figure 3.3: Illustration of Text Classifier used by GBS Crawler
Only C1 pages are considered targets of our crawler system. Previous studies have already shown the benefits of exploring links originating from targeted web pages (i.e., out-links) (Diligenti et al., 2000; Aggarwal et al., 2001; Chau and Chen, 2003; Chau and Chen, 2007). Accordingly, the queue management module in GBS assigns C1 pages’ out-links the highest weights. C2 pages are topically relevant but have irrelevant sentiment. For instance, if a company is interested in the amount of negativity surrounding a recent event, news articles describing the event (in an objective manner) would be considered C2 pages. C3 pages contain relevant sentiments but are not topically relevant. For instance, weblog and microblog pages often contain entries pertaining to an array of topics, which can diminish such pages’ overall relevance to any one topic (Thelwall, 2007). Using our company-event example, a blogger may express negative sentiments regarding the event in passing (e.g., with a single entry). C4 pages are those that are not considered relevant in terms of topic or sentiment. The weights for out-links of C2, C3, and C4 pages are calculated by the graph comparison module using their labelled web graphs, described in the following section.

3.3.2 Graph Comparison Module

Graph matching is the process of evaluating the structural similarity or dissimilarity of two graphs and a key task of structural pattern recognition. Two broad categories are exact graph matching, which requires a strict correspondence between two graphs or at least their subgraphs, and inexact graph matching, where a matching can occur even if there are some structural differences (Conte et al. 2004). Inexact graph matching has received additional attention in recent years since for many applications, exact matching
is impossible or computationally infeasible (Garey and Johnson, 1979; Conte et al., 2004).

One of the key characteristics of inexact graph matching is the similarity measure employed. Graph edit distance, which defines the matching cost based on costs of a set of graph edit operations (e.g., node insertion, node deletion, edge substitution, etc.), is considered one of the most flexible methods and has been applied to various types of graphs (Eshera and Fu, 1984; Baeza-Yates, 2000; Myers et al., 2000). However, existing methods for computing graph edit distance lack some of the formality and rigor associated with the computation of string edit distance. To convert graphs to string sequences so that string matching techniques can be used, Robles-Kelly and Hancock used a graph spectral seriation method to convert the adjacency matrix into a string or sequence order (Robles-Kelly and Hancock, 2005). For labelled graphs, random walk paths have been used to represent graphs as string sequences of node classes with associated occurrence probabilities (Kashima et al., 2003; Li et al., 2009). Accordingly, the graph comparison module utilized by GBS uses random walk paths to represent the web graphs associated with candidate pages as well as known relevant and irrelevant pages. Details regarding the graph comparison module are presented in the remainder of the section.

The graph comparison module analyses the labelled web graphs associated with pages deemed non-relevant by the text classifier during the crawling phase to determine if they are likely to lead to relevant pages. In other words, the objective of the graph comparison module is to determine whether “tunneling” through this particular non-relevant page could be fruitful. Algorithmically, the graph comparison module calculates
the weights of C2, C3, and C4 pages in the crawler’s queue based on the similarity of their discovered web graphs with those of training data, as illustrated in Figure 3.2.

The intuition behind the use of a graph-based tunneling mechanism is inspired by the observation that web graphs of irrelevant pages that lead to relevant content are subgraphs of relevant pages’ web graphs. Suppose the following path leads to a targeted C1 page: C1\(\rightarrow\)C2\(\rightarrow\)C3\(\rightarrow\)C1 (target), where the labels represent the classes associated with pages along the path. A focused crawler would explore all out-links of the seed C1 page and collect the C2 page. If it were a traditional topic-driven focused crawler, it would advance further along the path (since C2 is topic relevant) and collect the C3 page. Because this page is neither topic relevant nor sentiment relevant, the crawler would not be interested in exploring this path any further. Consequently, it would miss the targeted C1 page. To evaluate the value of irrelevant pages such as the C3 page from our example, the crawler cannot solely rely on its lexical content. However, let’s assume that the path that leads to C3 (C1\(\rightarrow\)C2\(\rightarrow\)C3) is also quite commonly found in the web graphs associated with C1 pages. This would suggest that analysing the out-links of the C3 page may lead to a C1 page. Hence, analysis of the similarity between web graphs of relevant and irrelevant pages may provide an estimate/indication of how close an irrelevant page is to relevant content.

As shown in the right side of Figure 3.2, initially we construct the web graphs of known relevant and irrelevant pages in the training dataset. A web graph, consisting of \(n\) levels of in and out links, is constructed for each page in the training data. The in-links are gathered using public in-link services such as Yahoo’s site explorer inbound links.
API. Due to computational limitations and efficiency issues, restrictions are imposed on the number of levels employed in the web graph, as well as the number of web pages (i.e., nodes) utilized per level. We set the level limit as 3 and sample 100 in-links for each node in the web graphs of the training data.

Nodes in the web graphs are labelled with their corresponding classes (C1-C4) using our text classifier module. Each web graph is then represented by various random walk path (RWP) sequences, where each RWP is comprised of a series of labelled nodes that signify the traversal of a particular path along the graph (Kashima et al., 2003). RWP sequences have been widely used in graph comparison tasks, for example patent classification using patent citation networks (Li et al., 2009). At each step, a random walk either jumps to one of the in-links or stops based on a probability distribution. Figure 3.4 represents a labelled web graph of page S. Nodes in the web graph are all ancestor pages of S and their class information is depicted by various node shapes (e.g., square, triangle, diamond, pentagon). Suppose we generate RWP sequences using a 0.1 stop/termination probability and equal “jump” probabilities, the highlighted RWP sequence S→C2→C1→C2 in the middle of the graph would have an occurrence probability of 0.3*0.3*0.9*0.1= 0.0081.
Figure 3.4: Random Walk Paths on a Labelled Web Graph of Page S

As illustrated in the left side of Figure 3.2, during the crawl, the graph comparison module is used to evaluate each C2, C3, and C4 page. The web graphs of irrelevant pages that our crawler finds in the crawling stage are constructed based on pages that have been already collected. While the maximum level of these web graphs may exceed 3, only nodes within the top 3 levels are used in the graph comparison module. GBS generates RWP sequences for the current set of collected irrelevant pages by following their in-links. Next, the web graphs of these candidate irrelevant pages are compared against those of pages in the training data.

The similarity between two graphs is measured by the aggregated value of similarities among their RWPs multiplied by these RWPs’ occurrence possibilities, and calculated using the following formula:

$$Sim\ (G, G') = \sum_h \sum_h' \cdot SimRwp(h, h')P(h|G)P(h'|G')$$  \hspace{0.25cm} (1)
where \( G \) and \( G' \) are two graphs, \( h \) and \( h' \) are RWPs of the two graphs, \( \text{SimRwp()} \) is used to calculate the similarity between RWPs, and \( P() \) returns the probability of each RWP in its graph.

As previously stated, the web graphs are comprised of the four types of nodes described in Section 3.1. Moreover, the RWP sequences used are also limited to 3 hops. Therefore, the types of RWP our graph module needs to deal with are predetermined: all possible permutations of C1-C4 nodes of length 3 or less (e.g., 211, 134, 231, 31). Hence, if we use “\( t \)” to represent one type of RWP, formula (1) can be transformed to:

\[
\text{Sim}(G, G') = \sum_t \sum_{t'} \text{SimRwp}(t, t')P(t|G)P(t'|G')
\]

where \( P(t|G) = \sum_h P(h|G) \text{Belong}(h, t) \), and \( \text{Belong}(h, t) = 1 \) if \( h \) belongs to type \( t \), 0 otherwise. If a type of RWP doesn’t appear in one graph, \( P(t|G) \) returns 0.

Since web graphs of candidate irrelevant pages are assumed as subgraphs of those of relevant pages, the two RWPs used in \( \text{SimRwp()} \) should be of different length. In other words, RWPs originating from the candidate irrelevant pages need to be shorter than those found in our training data. In order to compare such RWPs, we employ Levenshtein distance since it is well-suited for comparisons involving data of unequal size (Levenshtein, 1966). Levenshtein distance is a metric for measuring the amount of difference between two sequences (i.e., edit distance). The Levenshtein distance between two strings is given by the minimum number of operations needed to transform one string into the other, where an operation is an insertion, deletion, or substitution of a single character (Levenshtein, 1966). Therefore \( \text{SimRwp()} \) is replaced by \( LD() \) to represent the calculation of RWP similarity using Levenshtein distance.
If we use $SetG'$ to represent the set of web graphs $G'$ of our training data (either relevant set or irrelevant set), the web graph similarity of a candidate page with web graphs of a dataset can be calculated as an average similarity using the following formula:

$$Sim (G, SetG') = \frac{\sum G' \cdot Sim (G, G')}{\sum G'}$$

$$= \frac{\sum G' \cdot \sum_t \sum_{t'} LD(t, t')P(t|G)P(t'|G') \cdot \text{Short}(t, t')}{\sum t' \cdot P(t|G) \cdot \text{Training}(t, SetG') \cdot \sum_{t'} \sum_t LD(t, t')P(t'|G') \cdot \text{Short}(t, t')}$$

where $G$ is the web graph of a candidate page, $G'$ is that of a training page, $SetG'$ is the set of web graphs from the training data, $LD()$ calculates the Levenshtein distance of two RWPs, and $\text{Short}(a, b)$ returns 1 if path $a$ is shorter than path $b$ and 0 otherwise. $\text{Training}()$ represents all the calculations that are independent of $G$. These calculations return the possibility for a type of path “$t$” to appear in the web graphs of the training dataset and can be done in the training stage of our graph module. During the crawling stage, our crawler only needs to calculate $P(t|G)$, the possibility of a type of RWP in the discovered web graph of every candidate page. Such calculations are very fast considering the limited size of the web graphs and therefore the time complexity is definitely acceptable for crawlers.

The weight for out-links of a candidate page “$m$” is defined as the ratio of the page’s web graph similarity score for the relevant training data set to that for the irrelevant training data set:

$$Weight (m) = \frac{Sim (Gm, SetG\text{ Relevant})}{Sim (Gm, SetG \text{ Irrelevant})}$$
It is important to note that the web graph of a candidate URL can be updated during the crawling process when new ancestor pages (i.e., in-link pages) are discovered. Therefore the weights of candidate URLs should also be updated from time to time. In order to perform such updated in a computationally efficient manner, we update the weights of C2-C4 pages in the queue every time a predefined number of new irrelevant pages have been collected.

To the best of our knowledge, web graph similarity has not been explored in prior focused crawlers. There are two possible reasons: lack of information in the web graph structure and the time complexity issue. However, incorporating sentiment information into focused crawlers greatly enriches web graphs by providing an additional information dimension. The presence of additional node classes in the web graphs creates new opportunities for graph-based tunneling. Moreover, the time complexity for the graph comparison module utilized by GBS is computationally feasible due to the use of RWP-based inexact matching and training data that enables the use of a narrower set of promising web graph properties. In fact, recent machine learning studies have provided advanced methods to reduce the time complexity of string, tree, and graph-based matching to linear time (Rieck et al., 2010).

3.4 Evaluation

In order to examine the effectiveness of the proposed GBS crawler, which utilizes sentiment information and a labelled web graph, experiments were conducted that compared the system against traditional topic-driven crawlers, including Vector Space Model (VSM), Keyword-based method, Context Graph Model (CGM), Hopfield Net
(HFN), PageRank and Breadth-First-Search (BFS) (Brin and Page, 1998; Diligenti et al., 2000; Aggarwal et al., 2001; Chau and Chen, 2003; Chau and Chen, 2007). BFS was included since it is often used as a benchmark technique in focused crawling studies (Pant and Srinivasan, 2005; Chau and Chen, 2007). The other techniques incorporated are representative of those that adopt the aforementioned three types of contextual information: link context, ancestor pages, and web graph information (Pant and Srinivasan, 2005).

3.4.1 Test Bed and Training Data

Because of the dynamic nature of the Web (Arasu et al., 2001), we created a controlled environment for our experiments by taking a snapshot of a portion of the Web. Our test bed was built by collecting up to 6-levels of out-links from the homepages of 145 animal rights activist groups (e.g., Animal Liberation Front (ALF), People for the Ethical Treatment of Animals (PETA), etc.). These 145 homepages were also used as seed URLs for crawlers in the experiments. The test bed contained 524,483 web pages with a size of about 25 GB and included pages from websites, forums, and blogs.

We aimed to collect content containing negative sentiments towards organizations considered to infringe on animal rights and/or animal protection initiatives. Such content sheds light on an important and active constituency that exerts considerable influence on the political and corporate landscapes. The test bed also contained content with neutral or opposing sentiments, such as objective information and news about these groups, as well as criticism targeted towards animal rights activists by individuals and groups holding
opposing views. The variety of content in the test bed made it suitable for our experiments.

To train the text classifier and graph comparison modules of GBS, we built a training data set that consisted of 800 target/relevant web pages and 800 irrelevant ones. These pages were manually selected by two domain experts from both our test bed and the WWW. Consistent with prior work (Pant and Srinivasan, 2005), this data was used to train an accurate yet computationally expensive gold standard support vector machines (SVM) classifier that used over 10,000 learned attributes (Abbasi and Chen, 2008). The gold standard classifier’s attributes encompassed word n-grams, parts-of-speech tag n-grams, as well as various lexical and syntactic measures. The classifier attained 89.4% accuracy on 2,000 randomly selected testing pages from the test bed, which had been tagged by the two domain experts. This classifier was applied on the entire test bed to construct our gold standard. With an average run time of 3 seconds per page, the SVM classifier took nearly three weeks to process the entire test bed.
Figure 3.5: Test Bed Statistics by Level

Figure 3.5 shows a level-by-level breakdown of the number of relevant and irrelevant web pages in the test bed, based on the SVM classifier. Here, level 1 pages refer to the out-links of the 145 seed URLs, while level 2 pages are the level 1 pages’ out-links. The numbers displayed on each bar chart represent the number of relevant/irrelevant pages. For example at level 1, there are 2,776 relevant and 1,769 irrelevant pages, which are more than 60% of all the level 1 out-link pages. Not surprisingly, the percentage of relevant pages tends to decrease as we move further away from the seed URLs. This is why relevant pages in levels that are further out from the seed URLs pose difficulties for traditional focused crawlers; their successful collection often necessitates traversal of irrelevant in-link pages. In total, only 81,370 pages (15.5%) in the test bed were classified as relevant.

We used a public in-link service to collect up to 3 levels of in-link pages for the 1,600 training pages. The labelled web graphs of these training pages were used to learn
random walk path (RWP) sequences for our graph-based tunneling module. The in-link graph pages were labelled using the text classifier module described in Section 3.1, which assigned each page a label of C1-C4. As shown in formula (2), training of the graph module focuses on the function $Training()$, which returns the probability that a particular RWP sequence will appear in the web graphs associated with the training data. Since the in-link web graphs used by GBS did not exceed 3 levels, we only considered RWP sequences with a maximum length of 3 hops, excluding the target page. Since the graph module is only applied to irrelevant pages, only RWP sequences originating from irrelevant pages (i.e., ones labelled as C2, C3, or C4) were studied. This resulted in 60 possible RWP sequences.

Table 3.1 lists the top 20 RWP sequences based on the ratio of their probabilities of appearing in the web graphs of relevant training pages as compared to irrelevant ones. Each RWP sequence is represented by the labels corresponding to the graph nodes comprising that particular sequence. For example, RWP “211” refers to a path originating from a C2 page, and both the level 1 and level 2 in-link pages of this C2 page belong to C1. The number “5” is used to denote a sequence with an early termination. Hence, RWP sequences that end with the number “5” are RWPs with a length of two. For example RWP “215” is a two-node path in which a C2 page points to its ancestor C1 page. The example path shown in Figure 3.1 can be viewed as a successful 225 that leads to target content. Based on the results, RWP sequences that begin with “21” are pervasive at the top of the table. In other words, promising C2 pages are very likely to be outlinks of C1 pages (i.e., those considered relevant). Conversely, while the RWP sequence “411” has
the second highest relevant possibility, it is still not highly ranked due to the fact that its irrelevant possibility is also high. The last three shaded RWP sequences have a ratio less than 1, which suggests that they are more likely to link to irrelevant pages.

Table 3.1: Top 20 RWPs based on Graph Module Training

<table>
<thead>
<tr>
<th>RWP</th>
<th>Relevant Possibility</th>
<th>Irrelevant Possibility</th>
<th>Rel Pos / Irr Pos</th>
<th>RWP</th>
<th>Relevant Possibility</th>
<th>Irrelevant Possibility</th>
<th>Rel Pos / Irr Pos</th>
</tr>
</thead>
<tbody>
<tr>
<td>211</td>
<td>0.3716</td>
<td>0.1828</td>
<td>2.0327</td>
<td>225</td>
<td>0.1384</td>
<td>0.1027</td>
<td>1.3473</td>
</tr>
<tr>
<td>212</td>
<td>0.2779</td>
<td>0.1557</td>
<td>1.7845</td>
<td>222</td>
<td>0.1360</td>
<td>0.1032</td>
<td>1.3169</td>
</tr>
<tr>
<td>311</td>
<td>0.3181</td>
<td>0.1788</td>
<td>1.7793</td>
<td>411</td>
<td>0.3195</td>
<td>0.2533</td>
<td>1.2614</td>
</tr>
<tr>
<td>215</td>
<td>0.2752</td>
<td>0.1549</td>
<td>1.7772</td>
<td>331</td>
<td>0.1697</td>
<td>0.1426</td>
<td>1.1899</td>
</tr>
<tr>
<td>221</td>
<td>0.2583</td>
<td>0.1517</td>
<td>1.7028</td>
<td>313</td>
<td>0.1736</td>
<td>0.1467</td>
<td>1.1838</td>
</tr>
<tr>
<td>213</td>
<td>0.2274</td>
<td>0.1532</td>
<td>1.4846</td>
<td>214</td>
<td>0.2297</td>
<td>0.2199</td>
<td>1.0449</td>
</tr>
<tr>
<td>312</td>
<td>0.2270</td>
<td>0.1537</td>
<td>1.4768</td>
<td>412</td>
<td>0.2284</td>
<td>0.2276</td>
<td>1.0037</td>
</tr>
<tr>
<td>321</td>
<td>0.2238</td>
<td>0.1572</td>
<td>1.4240</td>
<td>232</td>
<td>0.1164</td>
<td>0.1187</td>
<td>0.9805</td>
</tr>
<tr>
<td>231</td>
<td>0.2136</td>
<td>0.1546</td>
<td>1.3820</td>
<td>421</td>
<td>0.2253</td>
<td>0.2318</td>
<td>0.9721</td>
</tr>
<tr>
<td>315</td>
<td>0.2040</td>
<td>0.1494</td>
<td>1.3653</td>
<td>241</td>
<td>0.2156</td>
<td>0.2239</td>
<td>0.9628</td>
</tr>
</tbody>
</table>

Of the 17 RWP sequences that have a ratio greater than 1, nearly two-thirds contain at least one C2 page. Analysis of the test bed reveals that two types of C2 pages are quite common in the RWP sequences. The first are news articles/reports from news websites such as CNN, MSNBC, and BBC. These pages usually describe stories and facts in an objective manner with neutral sentiment. The second are pages comprised of sentiments that oppose the opinions/views inherent in the relevant pages. Prior research has noted that the highly interactive nature of Web 2.0 media results in linkages between content
comprised of diverse and often opposing opinions (Tremayne et al., 2006). For examples, bloggers who argue with others often provide links to their opponents’ articles in their own blog entries in order to justify their arguments. Similarly, special interest groups often point to content associated with organizations and groups that do not share (or in some cases even oppose) their views, beliefs, and philosophies (Thelwall, 2007; Fu et al., 2010).

Equally interesting are the remaining one-third of the RWP sequences, which do not contain any C2 pages. These sequences are primarily anchored by C3 pages: ones that are not topically relevant but that have relevant sentiment. These are web pages that do not discuss the topic of interest in detail, but mention a few keywords in passing, with the appropriate sentiment polarity/intensity. Such sequences are important since a traditional topic crawler would have difficulty traversing C3 pages.

3.4.2 Experimental Setup

All comparison techniques were run using the best parameter settings, which were determined by tuning these methods’ parameters on the actual test bed data. Most of the parameter values used were consistent with prior research. The VSM and Keyword methods rely on link context for URL navigation (Menczer et al., 2004; Pant and Srinivasan, 2005). The two TF-IDF vectors for VSM contained all the words that appeared more than twice in the relevant and irrelevant training pages (Aggarwal et al., 2001). In contrast, the two vectors for the Keyword method contained fewer words since these words were selected using the information gain heuristic. For both VSM and Keyword, candidate URLs were weighted based on the ratio of cosine similarities
between their vectors and the vectors of relevant and irrelevant training data. For CGM, four Naive Bayes classifiers were constructed for targeted content and 3-level in-links (i.e., the context graph), respectively, using the training corpus (Diligenti et al., 2000). Every web page retrieved by CGM was represented by a reduced TF-IDF vector (relative to an identified vocabulary based on the training corpus), and was classified/assigned to a queue corresponding to the most probable layer of the context graph (i.e., target or level 1-3) based on the four classifiers’ predictions. For HFN, we followed the implementation employed by Chau and Chen (2003 and 2007) by using two set of key phrases selected using the information gain heuristic from web page content and anchor texts. The parameter values used in HFN’s spreading activation algorithm were similar to those suggested by Chau and Chen (2007). For PageRank, we set its damping factor to 0.9, consistent with previous studies (Menczer et al., 2004; Chau and Chen, 2007).

The evaluation metrics used to assess performance were F-measure, precision, and recall:

\[
F_{- \text{measure}} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
\]

where Precision = (number of relevant web pages retrieved)/(total number of web pages retrieved); Recall = (number of relevant web pages retrieved)/(total number of relevant pages available in the target set). Both recall and precision have been widely used in previous focused crawling studies (Menczer et al., 2004; Pant and Srinivasan, 2005).

In the following section, we describe the results for two experiments. In the first experiment, we evaluated the proposed GBS crawler in comparison with the six comparison methods: VSM, Keyword, CGM, HFN, BFS, and PageRank. All methods
were run using the seed URLs and test bed described in Section 4.1. In the second experiment, we conducted ablation testing to demonstrate the importance of the sentiment classifier and labelled graph-based tunneling module utilized by GBS to the methods overall effectiveness.

3.4.3 Experimental Results

Figure 3.6 shows the F-measure trends for GBS and the six comparison methods across the 528k web page test bed. The y-axis depicts the F-measure, while the x-axis displays the number of pages collected at that point in the crawl. Only four of the methods (GBS, CGM, BFS, and PageRank) traversed the entire collection. In contrast, HFN, Keyword, and VSM all used a stopping rule. GBS and CGM had the best overall performance. While these two techniques had similar F-measures on the first 50K pages, GBS performed considerably better than all comparison methods on the remainder of the pages, with F-measure values exceeding 50%. With respect to the remaining comparison methods, Keyword and BFS had the best performance, followed by VSM, HFN, and PageRank. PageRank’s poor performance is consistent with prior studies that have also noted that the method is less effective when applied to focused crawling tasks (Menczer et al., 2004; Chau and Chen, 2007). BFS performed well, with an average F-measure close to 0.28%. It possibly benefited from the fact that 65% relevant pages in the test bed were within the network of 3-level out-links, as shown in Figure 3.5. HFN stopped crawling at a very early state (about 70k pages) since it used a stopping rule that depended on the number of relevant phrases found in retrieved web pages’ body and anchor text (Chau and Chen, 2007).
Figure 3.6: F-Measure Trend for GBS and Comparison Methods

Figure 3.7 shows the precision and recall trends for GBS and the comparison methods. The x-axis displays the number of pages collected at that point in the crawl. The y-axis displays the precision (top of Figure 3.7) and recall (bottom of Figure 3.7). As previously described, Keyword, VSM, and HFN did not traverse the entire collection. Precision was computed as the percentage of collected pages that were relevant (Menczer et al., 2004; Pant and Srinivasan, 2005). Recall was computed as the percentage of total relevant pages collected at that point. Therefore, the recall values for all methods converged towards 100% as the total number of pages collected increased. The results reveal that the enhanced performance of GBS was balanced; it outperformed all comparison methods in terms of both precision and recall. With respect to the comparison
methods, the results were also consistent with CGM, Keyword, and BFS having the best precision and recall trends. For most techniques, precision decreased as the number of pages collected increased. This is not surprising since the proportion of relevant pages was greater in levels 1-2 of the test bed. Hence, as the crawlers went further out, their precision rates decreased since the number of potentially relevant pages subsided.

From the early onset, GBS had recall rates that were at least 10%-15% higher than the best comparison methods (CGM and Keyword), and 25%-30% greater than the next best methods: BFS and VSM. This performance gain has important implications for real-time business and marketing intelligence. GBS was able to collect a high proportion of the relevant pages much faster than the comparison methods. Case in point, GBS collected 50% of the relevant pages after traversing only 88k pages. In contrast, Keyword and BFS had to traverse 138k and 188k pages (i.e., 50k and 100k more pages) respectively, in order to reach 50% recall.

Table 3.2 shows the area under the curve (AUC) results corresponding to the F-measure, precision, and recall trends presented in Figures 3.6 and 3.7. Since three of the methods did not traverse the entire collection, the AUC values were standardized to a 0-1 scale by dividing them by the total number of pages collected. Based on the table, it is evident that GBS had the best AUC values for F-measure and recall. While Keyword performed marginally better in terms of its precision AUC value, this enhanced precision was coupled with significantly lower recall. The results presented in Table 3.2, along with Figures 3.6 and 3.7, suggest that GBS is well suited for focused crawling tasks involving topic and sentiment information.
Figure 3.7: Precision and Recall Trends for GBS and Comparison Methods
Table 3.2: Standardized Area Under the Curve (AUC) Values

<table>
<thead>
<tr>
<th>Technique</th>
<th>F-Measure</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>GBS</td>
<td>0.3857</td>
<td>0.3112</td>
<td>0.7841</td>
</tr>
<tr>
<td>CGM</td>
<td>0.3508</td>
<td>0.2850</td>
<td>0.7238</td>
</tr>
<tr>
<td>Keyword</td>
<td>0.3234</td>
<td>0.3300</td>
<td>0.4221</td>
</tr>
<tr>
<td>BFS</td>
<td>0.2689</td>
<td>0.2170</td>
<td>0.5779</td>
</tr>
<tr>
<td>PageRank</td>
<td>0.2218</td>
<td>0.1624</td>
<td>0.5201</td>
</tr>
<tr>
<td>VSM</td>
<td>0.2119</td>
<td>0.2157</td>
<td>0.3018</td>
</tr>
<tr>
<td>HFN</td>
<td>0.1145</td>
<td>0.2560</td>
<td>0.0847</td>
</tr>
</tbody>
</table>

We conducted level-based analysis to see how each method performed at different levels of the test bed (Diligenti et al., 2000). Since the seed URLs were considered level 0 pages, all out-links of the seed pages were considered level 1, while those pages’ out-links were level 2 (and so on). Figure 3.8 shows the recall trends for pages at levels 1-6. The results reveal that GBS performed well at all levels. It had the best recall values on levels 1, 3, 4, 5, and 6, while BFS had better performance on level 2. The enhanced recall of GBS on pages in deeper levels was a critical factor in its overall performance. The results support the notion that the graph-based tunneling mechanism and sentiment classifier utilized by GBS can improve focused crawling capabilities for tasks involving the collection of opinionated content on a particular topic.
3.4.4 Impact of Sentiment Information and Graph-based Tunneling

The experimental results demonstrate the effectiveness of the GBS crawler in sentiment-driven crawling tasks. We conducted further analysis to understand the
individual contribution of two important elements of GBS: the sentiment classifier and graph-based tunnelling mechanism. We performed ablation analysis where GBS was compared against two variations. The first was GBS without tunneling (GBS-T), in which the graph comparison module was not utilized. GBS-T only relied on the text classifier (described in Section 3.1) to assign relevance weights to pages. C2, C3, and C4 pages (i.e., those deemed irrelevant by the text classifier) were never moved up in the queue since there was no tunneling mechanism. The other variation was GBS without tunneling or sentiment information (GBS-TS). Like a traditional topical crawler, GBS-TS weighted all pages purely on the basis of topical relevance, using the topic classifier described in Section 3.1. The comparisons between GBS and GBS-T, and GBS-T and GBS-TS were intended to isolate the impacts of the labelled web graph based tunneling module and the sentiment classifier, respectively. The comparison between GBS and GBS-TS was designed to illustrate the collective impact of the tunneling module and sentiment classifier.

Figure 3.9 shows the F-measure, precision, and recall trends for GBS, GBS-T, and GBS-TS. All three settings performed comparably over the first 25k pages since the crawlers were primarily traversing the level 1 and 2 pages (as previously shown in Figure 3.8), which encompassed a large proportion of relevant pages. In other words, initially, GBS-T was able to perform well since its inability to tunnel was a non-factor, while GBS-TS was able to rely solely on topical relevance to attain decent results. However, from that point on, GBS separated itself from GBS-T and GBS-TS, with augmented f-measure, precision, and recall values. As the crawlers encountered a larger proportion of
irrelevant pages (i.e., pages from levels 3-6), the lack of tunneling in GBS-T and GBS-TS, as well as the absence of sentiment information in GBS-TS caused their performance to quickly deteriorate. The difference in performance between GBS and GBS-T, which was quite pronounced between 25k and 350k pages, demonstrates the usefulness of the graph-based tunneling mechanism. Similarly, the performance gain yielded by GBS-T over GBS-TS illustrates the utility of the sentiment classifier employed by GBS. Collectively, the results presented in Figure 3.9 underscore the effectiveness of two critical components of the GBS crawler.

In addition to improving collection precision and recall, GBS was designed to run in a computationally efficient manner. GBS was implemented in Java and run on a machine with an Intel Core 2 Duo 2.26 GHZ processor and 3GB of RAM using a maximum Java heap size of 1GB. By using random walk path based inexact graph matching, the graph-based tunnelling module incorporated by GBS was able to evaluate pages in a computationally efficient manner. The tunnelling mechanism had an average run time of 26.4 milliseconds per candidate page evaluated. The GBS crawler as a whole took under 3 hours to traverse the entire test bed, with an average crawl rate of 50 pages per second.
3.5 Conclusions

In this chapter, we proposed GBS, a focused crawler that uses a graph-based tunneling mechanism and a text classifier that utilizes topic and sentiment information. Two major contributions of our study are as follows. First, we demonstrated that sentiment information is useful for crawling tasks that involve consideration of content encompassing opinions about a particular topic. Second, we presented a novel graph-based method that ranks links associated with pages deemed irrelevant by utilizing labelled web graphs comprised of nodes labelled with topic and sentiment information.
This method helped GBS learn tunneling strategies for situations where relevant pages were near irrelevant ones. Collectively, these elements allowed GBS to outperform six comparison crawling methods in terms of F-measure, precision, and recall. For the majority of the crawl, GBS had recall rates that were at least 10% higher than the best comparison method. Moreover, GBS attained better recall rates at virtually all six levels. The experimental results suggest that GBS is able to collect a large proportion of relevant content after traversing fewer pages than existing topic-driven focused crawlers. Additionally, the graph-based tunneling module utilized by GBS is computationally efficient, making it suitable for “real-time” data collection and analysis. Overall, the findings support the notion that focused crawlers that incorporate sentiment information are well suited to support Web 2.0 business and marketing intelligence gathering efforts.
CHAPTER 4: EXPLORING GRAPH-BASED TUNNELING FOR FOCUSED CRAWLERS

4.1 Introduction

Two types of algorithms are critical for focused crawlers: Web analysis algorithms to judge the relevance and quality of retrieved Web pages and Web search algorithms to determine the order in which candidate URLs are visited (Qin et al., 2004). The most popular type of Web search algorithms are best-first search. In best-first search, retrieved pages are ranked by some heuristics and outlinks of the most promising page are chosen to be explored. Many such heuristics emphasize pages relevant to the targeted domain only so that the crawlers only search in directions originated from relevant pages. These crawlers are very effective for web communities in which relevant pages are closely linked with each other. However, as Qin et al. (2004) summarized in their study, there are three situations where pages relevant to a specific topic or domain are not closely linked with each other:

First, many pages in the same domain, especially domains where competition is involved, relate to each other through co-citation relationships instead of direct linkage (Dean and Henzinger, 1999). For example, major news websites often provide similar contents for a topic but these contents are rarely linked directly among these websites due to their competition.

Second, sometimes a group of relevant pages are linked by relevant pages from another website but they didn’t point back (Toyoda and Kitsuregawa, 2001). A bad set of starting URLs may lead the crawlers to miss one group of relevant pages.
Third, relevant Web pages could also be separated into different Web communities by irrelevant pages. Bergmark et al. (2002) studied 500,000 Web pages and found out that most pages relevant to the same target domain are separated by at least 1, to a maximum of 13, irrelevant pages.

In all the three situations, focused crawlers are very likely to be trapped in local optimal and miss other relevant content which are just a few steps away from collected pages. To address the above issue, researchers propose to use tunneling techniques which allow focused crawlers to traverse irrelevant pages in order to reach relevant ones (Martin et al., 2001). Bergmark et al. (2002) explored an adaptive tunneling technique which lets a crawler to continue search outlinks of an irrelevant page for a predefined number of steps. Diligenti et al. (2000) proposed the Context Graph Model which uses linguistic features of ancestor pages to predict how far away an irrelevant page is from a relevant target page. As illustrated in Chapter 3, web graphs rely the least on the lexical content of a page among the three categories of contextual information exploited by focused crawlers so that they are very suitable for tunneling. Several previous studies have pointed out that web graphs may provide essential cues about the merit of following a particular URL, resulting in improved tunneling capabilities (Broder et al., 2000; Pant and Srinivasan, 2005). However, previous researchers have seldom explored web graphs due to limitations in available graph information and computational constraints.

With the help of sentiment information, I have proposed a random walk based graph tunneling techniques for focused crawlers, described in Section 3.3.2. The results presented in Section 3.4.4 demonstrate the usefulness of labeled web graph similarities in
tunneling. However, random walk based methods suffer from high time complexities and do not scale well with large graphs. To address this issue, most computation burden was shifted to the training stage by identifying calculations independent of the page to be evaluated. Such approach resulted in a very long training time and still didn’t solve the scalability issue.

In this chapter, I further extend the work in Chapter 3 by exploring the possibilities of using other graph comparison techniques in tunneling for focused crawlers. We aim to find techniques that allow fast training and scale up to large graphs. Subtree-based methods are selected to be explored based on our literature review and a simple binary subtree based tunneling algorithm is proposed and evaluated. Experiment results demonstrate that subtrees are effective substructures of graphs to be used in tunneling and applicable to large graphs.

The remainder of this chapter is organized as follows: Section 4.2 presents a review on state-of-the-art graph kernels and discusses their possibilities to be used in tunneling. Based on the idea of subtree kernels, a simple binary subtree tunneling algorithm is proposed in Section 4.3. Section 4.4 describes a preliminary experiment to evaluate subtree based tunneling using the proposed algorithm. Section 4.5 presents concluding remarks.

4.2 Literature Review

Graph comparison has been widely studies in many areas such as chemistry, bioinformatics, and sociology. Shervashidze et al. (2009) categorized existing graph comparison techniques into three categories: set based, frequent subgraph based, and
kernel based. Set based methods represent graphs as set of nodes or edges and measure their similarities. They neglect the structure of the graph, i.e. their topology, so that are not very effective in graph comparison. Frequent subgraph based algorithms identify discriminative subgraphs using feature selection techniques (e.g., Yan and Han, 2003). They respect graph topology but the complexity grows exponentially when graph size is increased. Kernel based approaches are most popular in recent years because they represent a balance in computational complexity and topology exploitation. Kernel methods can be applied in high dimensional feature spaces without suffering from the high cost of explicitly computing the feature map. The rest of this section focuses on graph kernels and discuss the possibility for various kernels to be used in focused crawler tunneling.

4.2.1 Graph Kernels

The general idea of graph kernels is to measure common subgraphs of two graphs (Haussler, 1999). Current state-of-the-art graph kernels can be categorized into three classes: graph kernels based on walks and paths, graph kernels based on subtree patterns, and graph kernels based on limited-size subgraphs (Shervashidze and Borgwardt, 2009).

Paths are sequences of unique nodes. As pointed by Borgwardt and Kriegel (2005), all path kernel which compares all the paths pairwise and longest path kernel are both NP-hard to compute. However, shortest path kernel is possible since it is computable in cubic time by the classic Floyd-Warshall algorithm (Floyd, R., 1962; Warshall, S., 1962). Consequently, they proposed a shortest path kernel which counts pairs of nodes labeled with identical shortest path distance. This shortest path kernel performs well with graph
of small size but takes very long time for graph of large size (Shervashidze and Borgwardt, 2009).

Walks are sequences of nodes that allow repetitions of nodes. Random walk kernel is based on the idea to count the number of matching random walks in two input graphs (Gärtner et al., 2003; Li et al., 2009). At each step, a random walk either jumps to one of the in-links or stops based on a probability distribution, as illustrated in Figure 3.4. This type of kernel is further improved by measuring similarities between walks which are not identical (Kashima et al., 2003). However, the computational complexity of random walk kernels is high due to the fact that all pairs of random walks need to be compared. Although fast kernel computation has been developed for random walk kernels to reduce the computational time to cubic (Vishwanathan et al., 2007), this kernel is still much slower than other state-of-the-art graph kernels, demonstrated in Shervashidze and Borgwardt’s study (2009). Besides, walk kernels also suffer from the problem of “tottering”, i.e., by iteratively visiting the same cycle of nodes, a walk can generate artificially high similarity values. In comparison, shortest path kernels have no tottering.

The graph comparison module proposed in Chapter 3 is based on the idea of random walk kernel. To address the computational complexity issue, intensive training was conducted in the training dataset, described in Section 3.3.2. The level limit of graph was set to 3 and a maximum of 100 inlinks were sampled for each page to be considered in the calculation. The total number of walk types was also small since there are only four labels in graphs explored in that study. As a result, the graph comparison module only needs to summarize the possibilities of each random walk type in the evaluated web
graph during the crawling state and the similarity calculation is very fast. However, the training is still very time-consuming.

Another limitation of walk kernels is that different graphs can be mapped to identical points in walks feature space, illustrated in Figure 4.1 and Figure 4.2 (adopted from Ramon and Gärtner, 2003).

![Figure 4.1: Directed Graphs Mapped to the Same Point in Walks Feature Space](image1)

![Figure 4.2: Undirected Graphs Mapped to the Same Point in Walks Feature Space](image2)

Subtree kernels which compare tree-like substructures of graphs are proposed to address this limitation. They may distinguish between substructures that walk kernels deem identical. The first subtree kernel was defined by Ramon and Gärtner (2003). It compares all pairs of nodes from two input graphs by iteratively comparing their neighborhoods and counts the number of subtree pairs of same pattern with a tree height limitation. Decay factors are also included to cause higher trees to have a smaller weight.
in the overall sum. This type of kernel has been further refined to consider unbalanced subtree and k-ary subtree with at most k children per node and avoid tottering (Mahé and Vert, 2006; Bach 2008). The complexity concern of subtree kernels is successfully addressed by Shervashidze and Borgwardt (2009) by adopting the Weisfeiler-Lehman test of isomorphism (Weisfeiler and Lehman, 1968). For two graphs with n nodes and m edges and maximum degree d, these kernels comparing subtrees of height h can be computed in O(mh), Their subtree kernels are most accurate and scales up to large, labeled graphs compared with other graph kernels based on their experiment.

Kernels based on limited-size subgraphs are proposed by Shervashidze et al. (2009). Their kernel is based on the distribution of subgraphs of size k (k= 3, 4, 5), which they refer to as graphlets. This kernel is fastest among other graph kernels for small to middle size graphs but of low accuracy. For large size graphs, it is comparable to the subtree kernel but slower in run time.

4.2.2 Graph-based Tunneling for Focused Crawler

As described in Chapter 3, our sentiment focused crawler uses a text classifier to classify web pages into four classes based on their topic and sentiment relevance. Using such class information, labeled web graphs of retrieved pages can be generated, thus providing an opportunity to explore graph-based tunneling. The intuition behind the use of a graph-based tunneling mechanism is inspired by the observation that web graphs of irrelevant pages that lead to relevant content are subgraphs of relevant pages’ web graphs. However, several criteria need to be met in order for graph comparison algorithms to be effective in tunneling based on our experience in Chapter 3.
First, graph comparison algorithms need to be efficient in terms of both running time and accuracy, because focused crawlers process a very large number of web pages in a short time. Moreover web graphs are dynamic during the crawling due to new discovered inlinks so that the algorithms have to update the weights of candidate pages frequently in a reasonable time.

Second, the algorithms must be scale up to large graphs. The web graph of retrieved pages can be as simple as a single path and as complex as those with hundreds of nodes. Algorithms that perform badly on large graph cannot handle the tunneling task in focused crawlers.

Third, since web graphs to be compared in focused crawler are all rooted from pages to be evaluated, comparison algorithms should focus on substructures rooted from these pages or close neighbors of these roots instead of comparing all pairs of nodes in two graphs.

Based on these criteria, we evaluate the possibilities for the abovementioned state-of-the-art graph kernels to be used in tunneling. The shortest path kernel is first excluded. Its run time for large graph is pretty long. Besides, if it only utilizes shortest paths related to rooted pages, its performance is likely to be much worse due to the few types of shortest path available. Random walk kernels are the slowest according to Shervashidze and Borgwardt’s study (2009). However, the GBS algorithm we proposed in Chapter 3 limits the size of graphs and shifts the computation burden to the training stage, resulting in an acceptable random walk based graph comparison module. As to subtree kernels and subgraph kernels, they both meet the first two criteria. But unlike walks and subtrees,
graphlets do not have a definition of roots so that the graphlet subgraph kernel fails the third criteria.

Subtree kernels have the best performance among the four types of graph kernels in general. The substructures they adopt match the nature of web graphs very well: single root and similar parent-children relationship. The recent improvement in fast kernel computation also demonstrates their abilities for large graph comparison. Consequently, they are very suitable to be used in graph-based tunneling for focused crawlers.

4.3 Research Design

The discussion of previous section clearly shows that subtree kernels are promising in tunneling for focused crawlers. In this exploratory study, we proposed a simple subtree-based graph matching algorithm based on the ideas of subtree kernels. The algorithm only considers the most basic binary subtrees, i.e., 2-ary subtree, with a size limitation of 3 nodes and no specific height restrictions. We refer the algorithm as Subtree-2a3n0h. The binary subtree can be further divided into balanced binary subtree whose height is 2 and unbalanced binary subtree whose height is 3 and is identical to walk of length 3. Since the graph module is only applied to irrelevant pages and there are 4 class labels in the web graph, there are totally 48 unbalanced subtree patterns and 30 balanced ones (children nodes are not differentiated by left or right). These 78 patterns construct an ordered set of substructures for graph representation.

Following what previous studies did (Shervashidze and Borgwardt, 2009; Shervashidze et al., 2009), each graph \( G \) is represented by a normalized vector \( F_G \) whose i-th component corresponds to the percentage of occurrence of the i-th subtree pattern.
Using normalized percentage value instead of frequencies is to account for differences in the sizes of the graphs. Given two graphs $G$ and $G'$, their similarity score is calculated as follows:

$$\text{Sim}(G, G') = F_G^T \cdot F_{G'}$$

To be consistent with the random walk module, the weight of a candidate irrelevant page is still the ratio of the page’s web graph similarity score for the relevant training data set to that for the irrelevant training data set. Both data sets are represented as a single aggregated vector for fast computation.

We need to make sure the proposed Subtree-2a3n0h algorithm meets the three criteria described in Section 4.2.2. The performance of this algorithm will be evaluated in the next section. As to the run time complexity, the algorithm is in fact linear to the size of the partial web graph it can accessible. The unbalanced subtrees are identical to walks so that a traversal of the graph is enough to count their frequencies. The balanced ones can also be easily counted by mathematical combination once the inlink class distribution is counted during graph traversal. Therefore the proposed algorithm does not violate the first two criteria. Its fast computable nature also allows us to train the algorithm using large graphs which do not limit the number of pages at each inlink level.

To meet the third criterion, only subtrees rooted from the candidate pages are counted during the crawling. Besides, when calculating the aggregated vectors for training data sets, subtrees rooted from nodes that are closer to level 0 training pages should have a higher weight in the final total counts. Inspired by the decay factor used in subtree kernels (Ramon and Gärtner, 2003), our algorithm also adopts a positive decay
factor whose value is smaller than 1 to cause farther subtrees to have smaller weights in the overall sum. For example, a decay factor of 0.9 means the counting result at level $k$ inlinks is of 90% weight in the total sum compared with result at level $k-1$ inlinks.

4.4 Evaluation

To evaluate the proposed Subtree-2a3n0h algorithm, we replaced the random walk based graph module with this algorithm in our GBS crawlers and compare it with the original GBS proposed in Chapter 3. Context Graph Model (CGM) is also included in the comparison for two reasons. First, it performed best among all the comparison techniques in the experiments of Chapter 3. Second, it uses a text-based tunneling algorithm by building Naïve Bayes classifiers for each layer of the relevant training data’s context graph. These classifiers are then used to predict how far away an irrelevant page is from a relevant target page.

To analyze whether the proposed decay factor will affect the performance, two versions of Subtree-2a3n0h with decay factor 0.9 and 0.0 were compared. Decay factor 0 means only subtrees rooted at level 1 inlinks of training data set are counted. Decay factor 0.9 is consistent with the jumping possibility of random walk graph module in Chapter 3. The same test bed and training data described in Chapter 3 are used for the experiments. Evaluation measures are still F-measure, precision, and recall so that we can easily compare Subtree-2a3n0h with other algorithms including the random walk based GBS explored in Chapter 3.

Figure 4.3 shows the F-measure, recall, and precision trends for GBS proposed in Chapter 3 (“RandomWalk”), GBS using Subtree-2a3n0h with 0.9 decay factor
(“Subtree_D09”), GBS using Subtree-2a3n0h with 0 decay factor (“Subtree_D00”), and CGM across the 528k web page test bed. The y-axes depict the F-measure (top of Figure 4.3), recall (bottom-left of Figure 4.3), and precision (bottom-right of Figure 4.3) while the x-axes display the number of pages collected at that point in the crawl.

Figure 4.3: F-Measure, Recall, and Precision Trends for RandomWalk, Subtree_D00, Subtree_D09, and CGM
RandomWalk performed best in all three evaluation measures. Subtree_09 ranked second and is competitive to RandomWalk. What’s interesting is that Subtree_09 showed similar trend shapes to RandomWalk in evaluation. This means both methods are effective in capturing the web graph information of candidate pages. In comparison, Subtree_D00 didn’t perform well in the experiment. It was even worse than CGM which doesn’t utilize sentiment information. It demonstrates that the decay factor plays an important role in subtree-based graph comparison methods. Further exploration on the decay factor is needed to find out an optimal value for this parameter.

Table 4.1 shows the area under the curve (AUC) results corresponding to the F-measure, precision, and recall trends presented in Figures 4.3 for the first 200k web pages collected. After 200k pages, the percentage of relevant pages left is very small so that all tunneling methods performed similarly. Based on this table, it is evident that both RandomWalk and Subtree_D09 are effective in tunneling for sentimental crawlers.

Table 4.1: Standardized Area Under the Curve (AUC) Values for First 200k Pages

<table>
<thead>
<tr>
<th>Technique</th>
<th>F-Measure</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>RandomWalk</td>
<td>0.4252</td>
<td>0.4486</td>
<td>0.5119</td>
</tr>
<tr>
<td>Subtree_D09</td>
<td>0.3997</td>
<td>0.4267</td>
<td>0.4793</td>
</tr>
<tr>
<td>Subtree_D00</td>
<td>0.3522</td>
<td>0.3756</td>
<td>0.4311</td>
</tr>
<tr>
<td>CGM</td>
<td>0.3741</td>
<td>0.4061</td>
<td>0.4479</td>
</tr>
</tbody>
</table>

As to the training time, RandomWalk took more than 12 hours under the limitation of 100 inlinks for each node in the web graphs on a machine with an Intel Core 2 Duo 2.26 GHZ processor and 3GB of RAM using a maximum Java heap size of 1GB. In
comparison, subtree methods’ training was about 100 times faster and only took 7 minutes if the same inlink limitation was applied. Since subtree methods scale up to large graphs, both Subtree_D09 and Subtree_D00 were trained without any limitation on the number of inlinks and it only took them 2 hours, which was still much faster than random walk training on restricted graphs. For a particular large web graph consisting of 11,041 unique nodes and almost a billion nodes to traverse due to bidirectional links and tottering, the proposed subtree methods took half an hour to count its subtree frequencies, while it would take the random walk based method about 50 hours. This confirms subtree methods’ advantage in exploring large graphs. During the crawling stage, both two types of methods took about 3 hours to traverse the entire test bed under the same inlink number limitation.

In sum, compared with the random walk based tunneling method, the proposed subtree-based ones performed worse in F-measure, precision and recall but better in scalability and training time. By adopting a decay factor which causes farther subtrees to have smaller weights in the overall sum during training, subtree methods’ performance was greatly improved and close to the random walk one.

4.5 Conclusion

In this chapter, we extended the work in Chapter 3 to further explore graph-based tunneling in focused crawlers in order to find techniques that overcome shortcomings of random walk based tunneling by being able to scale up to large graphs and allowing fast training. We reviewed several types of state-of-the-art graph kernels that utilize substructures from simple ones like walks and paths to complex comes including trees
and subgraphs. Based on runtime requirements of focused crawlers and the natural of web graphs to be compared, we discussed the possibilities for those graph comparison algorithms to be applied in tunneling and identified tree-based graph kernel as a candidate technique. In a preliminary experiment, we compared a simple subtree-based graph tunneling algorithm with the random walk one proposed in Chapter 3 and CGM which uses a text-based tunneling algorithm. The experiment results showed that the proposed simple subtree methods are good at analyzing large graphs and run much faster in training. Although their performance in F-measure, precision, and recall was worse than random walk tunneling, the difference was small when a decay factor was applied in training.

It is noticed that the proposed subtree methods are simple and have a lot of room for improvement. Many parameters of subtree patterns, such as number of children per node, total number of node, maximum height, and the decay factor, have not been tuned. Techniques that avoid tottering can also be applied. For more complex tree patterns, fast subtree kernel computation method can be used to keep the speed advantage.

The graph comparison methods reviewed and developed in this chapter are not limited to data collection tasks in our CSI framework. They can also facilitate data investigation tasks by applying them to other types of networks, as long as graph similarity is meaningful for the specific task. For example, interaction networks of successful open source projects can help us identify promising ongoing projects by comparing the network similarities using the methods described in this chapter. Similar
approach can be applied to evaluate the quality of wiki content. We are interested in these
directions and plan to explore them in the future.
CHAPTER 5: TEXT-BASED VIDEO CONTENT CLASSIFICATION FOR ONLINE VIDEO-SHARING SITES

5.1 Introduction

User behavior in Web 2.0 communities has changed from just browsing web pages to generate and spread their own content and ideas. To obtain insight from user-generated information, the ability to collect and analyze the considerable quantity of information becomes a challenge. Classification technologies provide promising methods to organize data according to different perspectives. Many studies have used classification technologies to analyze text-based data collected from blogs and forums and obtain insights. For example, Abbasi et al. (2008) applied sentiment analysis to improve opinion classification of web forums in multiple languages. Zheng et al. (2006) adopted writing style features to identify online authorship.

Like blogs and forums, video-sharing websites are an important part of Web 2.0. For example, YouTube, the world’s largest video-sharing website, receives more than 65,000 videos and 100 million video views every day. Video classification techniques can be used to improve user experiences with video-sharing websites by identifying videos more closely related to users’ personal interests and distinguishing them from the many irrelevant videos that are obtained by using keyword searches alone.

Another challenging issue for Web 2.0 sites is the issue of illegal content such as child pornography or threatening content such as sites exhorting violence and extremism. Among these, violent extremism content is considered to be among the most dangerous, especially after the attack of September 11th. The U.S. government invests many
resources in detecting potential terrorism and protecting the United States from extremist violence. Chen et al. (2008) found that extremists use Web 2.0 as an effective platform to share resources, promote their ideas, and communicate among each other. For now, YouTube provides only the “flag” mechanism for users to mark inappropriate videos (Chen et al., 2008). Video classification can help video-sharing Web sites automatically manage videos by classifying illegal or offensive videos and distinguishing them from acceptable ones. Moreover, accurate video classification results are very useful for identifying implicit cyber communities on video-sharing Web sites (Kumar et al., 1999).

Implicit cyber communities can be defined only by the interactions among users, such as subscription, linking, or commenting. Chau and Xu (2007) studied implicit cyber communities for blogs while Fu et al. (2008) used interaction-coherence information to identify user communities for Web forums. However, few studies have addressed the cyber communities on video-sharing Web sites.

Different from the studies of forums and blogs which used text features to represent collected data, most studies in video analysis have used non-text features extracted from video clips and audio tracks (e.g., Messina et al., 2006). However, video-sharing communities not only allow users to upload and share videos, but also provide functions to enable users to interact with other users, which generate additional text information. For instance, YouTube allows its users to comment on and rate videos, create personal video collections, and categorize and tag videos they upload. Such user-generated text information may contain explicit information related to video content and hence can be used to classify videos. In addition, this information can be easily obtained by parsing
web pages or using various Web APIs (Chen et al., 2008). But for now, few studies have explored user-generated text features in video classification.

In order to make use of the information provided by user-generated data and evaluate their effectiveness in online video classification, we propose a framework of video classification for video-sharing websites by using user-generated text data such as comments, descriptions, video titles, etc. We evaluated the performance of different classification techniques and text feature sets. In addition, we conducted key feature analysis to identify the most useful user-generated data for online video classification and showed how our framework can help identify implicit cyber communities on video-sharing websites.

While Chapters 2, 3, and 4 show improvements we have made for data collection task in the CSI framework, this chapter presents our effort in the data selection task. We focus on a specific Web 2.0 medium, online video sharing sites, and make our contribution to the CSI framework by creatively using text features extracted from user-generated text content for online video classification.

The remainder of this chapter is organized as follows. Section 5.2 presents a review of current video classification research. Section 5.3 describes research gaps and questions, while Section 5.4 shows our research design. The testbed created and used in our experiment is discussed in Section 5.5. Experiments used to evaluate the effectiveness of the proposed approach and discussions of the results are illustrated in Section 5.6. A case study showing how the proposed framework can help identify implicit cyber communities on video-sharing websites is presented in Section 5.7. Section
5.8 concludes with closing remarks and future directions.

5.2 Literature Review

Among all data types, such as text, audio, and image, video has the highest capacity in terms of the volume and the richness of the content. Videos not only contain diverse data types, i.e., image, audio, and text data, but also combine these data types together and further create deeper semantic meanings. These semantic meanings and information can be easily recognized by human beings, but how to leverage information technologies to process videos and extract these semantic meanings automatically is a challenging issue.

Semantic gap, referring to the gap between video features (e.g., color, texture, and volume of audio) and semantic concepts (Lew et al., 2006,) which are concepts meaningful to human beings (e.g., cars, faces, buildings, etc.,) is one of the most challenging issues of video classification studies. To bridge semantic gap and obtain better understanding of video contents, many different techniques have been developed to enhance classification accuracy (accuracy refers to the percentage of correctly classified instances), and different video features have been identified to represent videos better (Dimitrova et al., 2000; Hsu and Chang, 2005; Hung et al., 2007). Common video classification research characteristics include domains, feature types, and classification techniques. Table 5.1 shows the taxonomy of these important video classification analysis characteristics. The taxonomy and related studies are discussed in detail below.
### Table 5.1: Taxonomy of Video Classification Studies

<table>
<thead>
<tr>
<th>Domain</th>
<th>Description</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>General TV program</td>
<td>Sport games, News, Weather reports, Commercials</td>
<td>D1</td>
</tr>
<tr>
<td>Movie and Movie Preview</td>
<td>Movies, Movie preview videos</td>
<td>D2</td>
</tr>
<tr>
<td>Specific Scenario Video</td>
<td>Staff meeting videos</td>
<td>D3</td>
</tr>
<tr>
<td>Archival Video</td>
<td>Videos of TRECVID, Internet Archive, or Open Video</td>
<td>D4</td>
</tr>
<tr>
<td>Video-sharing Website Video</td>
<td>YouTube, MySpace, and Flicker videos</td>
<td>D5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Feature Types</th>
<th>Description</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-text Features</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Low-level Video Features</td>
<td>Non-text features extracted from row clips, such as color, motion and texture features</td>
<td>NT-L</td>
</tr>
<tr>
<td>Mid-level/High-level Video Features</td>
<td>Semantic features, such as face, object, and anchor detection</td>
<td>NT-MH</td>
</tr>
<tr>
<td>Text Features</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Video Embedded Text Features</td>
<td>Text features from video embedded information, such as subtitles and close-caption</td>
<td>T-E</td>
</tr>
<tr>
<td>User-generated Text Features</td>
<td>Text features from user-generated information, such as video titles, descriptions, tags and category names</td>
<td>T-U</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Techniques</th>
<th>Description</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Machine Learning Techniques for Classification</td>
<td>Such as Hidden Markov Models (HMM), Gaussian Mixture Model (GMM), and Support Vector Machine (SVM)</td>
<td>T1</td>
</tr>
</tbody>
</table>

### 5.2.1 Video Domains

There are five main categories of video domains: general TV programs, movies and movie previews, specific scenario videos, archival videos, and video-sharing website videos. Besides the basic components of videos, i.e., image and audio, videos within some domains provide extra information which can be utilized to classify videos more
accurately. For example, some general TV programs contain subtitles and closed-captioning which can be extracted to help understand video contents.

TV programs are the most traditional video resources and therefore most studies have used general TV programs as their experiment data. Some studies in this domain have classified TV programs according to program types, such as news, sports games, weather reports, and commercial advertisements. Montagnuolo and Messina (2007) classified 700 broadcasted programs into seven TV program types, and reported 86.2% average precision (average precision refers to the average percentage of correctly classified instances, which are programs in this case, across all predicted classes). Other studies focused on classifying a single type of TV programs into different specific events. For example, Hung et al. (2007) classified baseball videos into several important events, such as homerun, hit, strike-out, etc., and achieved 95% average precision.

The second domain is movies and movie previews. Movies play an important role in the entertainment industry. Approximately 4500 films (about 9000 hours of video) are produced every year (Rasheed et al., 2003). Hence, some studies have focused on classifying movies according to their genres. Vasconcelos and Lippman (2000) classified movies into three categories: romance/comedy, action, and others, including horror, drama, and adventure. In addition to movies, movie previews, previews of upcoming movies or previews provided by DVD rental companies, have also been used as testbeds in previous studies. For instance, Rasheed et al. (2003) classified movie previews into different categories such as comedies, action films, dramas, and horror films.

Specific scenario videos are videos generated by individuals for specific events, such
as meeting or lecture videos. For example, business meetings captured in video were used by Girgensohn and Foote (1999) to classify them into presenter, slides, and audience scenes.

Another important video domain is archival videos which are generally collected and provided by organizations (e.g., Internet Archive). These videos are collected from various media sources (such as movies, TV programs, and personal-made videos) and will be well-organized into different categories (e.g., such as cartoons, movies, news, etc.) before providing to publics. Some organizations will further provide pre-processed videos for researchers to perform their experiments. For example, TRECVID was founded in 2003 and is now a well-known workshop that provides large testing datasets scored by uniform scoring procedures for video information retrieval studies.

As Web 2.0 gains in popularity, the study of video-sharing websites has become an emerging domain of interest. Videos are uploaded by online users and reviewed by the public. Video sharing websites, such as YouTube and Yahoo Video, usually provide a convenient environment for users to discuss and comment on videos. Some sites even provide APIs that allow people to easily extract relevant information from videos of interest.

Consequently, videos from video-sharing websites contain several unique characteristics. First, most online videos are short, and their contents are highly diverse. Second, much user-generated data, such as descriptions and comments, can be collected easily for each online video. Information about video authors and reviewers is sometimes available, including other videos uploaded by the same person, etc. Third, due to
copyright issues, online videos on video-sharing sites may not be always available for people to download and analyze. Hence, applying non-text video features to online videos classification may have difficulties in collecting training datasets. User-generated data can be used as an alternative, because they can be easily obtained and sometimes contain more explicit information about the content of associated videos. Currently, few studies have emphasized on such an approach.

5.2.2 Feature Types

Features used in video classification studies can be divided into two main categories, non-text features and text features. While non-text features can be further split into low-level video features and semantic video features, text features contain text features extracted from videos and user-generated text features.

5.2.2.1 Non-Text Features

Non-text features are features extracted from the two basic components of videos, audio and image. Djeraba (2002) stated that low-level video features are features extracted from the video clips and audio tracks without referring to any external knowledge. For example, color, texture, and motion are major low-level features extracted from video clips (Gibert et al., 2003; Huang et al., 1999; Ma and Zhang, 2003). Fischer et al. (1995) utilized audio features such as volume of audio, audio wave forms, and audio frequency spectrum. Other features such as edge, lighting, and shot length were also adopted in some studies (e.g., Rasheed et al., 2003). Moreover, the text trajectory feature, which refers to the motion of texts in continuous video clips, is considered to be a low-level non-text feature as opposed to a text feature (e.g., Dimitrova et al., 2000).
Zhou et al. (2000) and Luo and Boutell (2005) claimed that low-level video features lack the capacity to identify semantic concepts, which make them inefficient to use for video classification alone. To solve this problem, mid-level and high-level video features are proposed to bridge the “semantic gap” (Lew et al., 2006), the gap between low-level video features and semantic concepts (Hsu and Chang, 2005). These two feature types are generated from low-level features and are also known as semantic features.

Mid-level video features are extracted by mid-level feature detectors or sensors, which are pre-trained classifiers used to capture mid-level features from input data, and each of them represents an atomic semantic concept, which cannot be represented by combinations of other semantic concepts. Some examples used in previous studies include cityscape, landscape, face, object, indoor, outdoor, etc. (Chellappa et al., 1995; Lin and Hauptmann, 2002; Samal and Iyengar, 1992). Xu and Chang (2008) developed 374 mid-level feature detectors to detect video events. The average precision for event detection was between 24.4% and 38.2%. Mid-level features have been adopted in many studies. Dimitrova et al. (2000) used text and face trajectories to classify videos into four categories (i.e., news, commercials, sitcoms, and soaps) and reported 80% accuracy.

High-level video features are features containing multiple semantic concepts, which generally require human to define (Borgne et al., 2007). Some studies relied on domain knowledge to achieve high-level analysis. Duan et al. (2003) combined sport domain knowledge with mid-level features to conduct high-level video analysis to categorize segments of videos of field-ball sports into different events. For example, Duan et al. (2003) constructed several mid-level feature detectors to capture semantic shots (such as
field view, audience, goal view, player following, and replay) from videos of soccer games. With the help of sport domain knowledge, they first defined “in play segments”, video segments consisting of shots taken when a game is playing (e.g., field views and player following,) and “out of play segments”, video segments containing shots taken when a game has been stopped by referee (e.g., audience and replay). Further, specific events of each segment were identified. For example, kickoff, passing, and shot were captured from “in play sections”, while penalty kick, throw-in, and corner kick were identified from “out of play segments”.

5.2.2.2 Text Features

In addition to non-text features, some studies adopted text features to enhance the classification performances. Subtitles and closed-captions are the typical text information that can be extracted from videos of various types, such as TV programs and movies. Lin and Hauptmann (2002) extracted closed-captions from CNN broadcasts and treated each word of the closed-captions as a feature. Bag-of-words was used to represent the broadcasts and their experiment results demonstrated that text features can improve the precision of classification results.

In addition, text information can also be obtained from audio tracks using speech recognition techniques (Smoliar and HongJiang, 1994). For example, Amir et al. (2004) transcribed audios recordings, generated a continuous stream of timed words, and included the text information for video event detection.

User-generated text information is a new text data source emerging only recently with video-sharing website videos. Different from the other four video classification domains,
online videos are generally shorter but contain more user-generated information. In the Web2.0 architecture of participation, online users not only review videos, but also comment on videos and exchange opinions with other reviewers. Through the user-participation process, much video-related text data are created. These data often contain explicit information about the video content and can be utilized to classify videos. In addition, more and more user-generated text information can be easily collected from video-sharing websites. For example, the YouTube API allows users to extract information such as titles, user comments, descriptions, tags, etc (Chen et al., 2008).

Sharma and Elidrisi (2008) recently used video tag information to classify YouTube videos into YouTube pre-defined categories such as education and comedy. They claimed that video tags are given by users and therefore contain highly user-centric information and can be used as the meta-data of videos. Their results achieved around 65% accuracy. To the best of our knowledge, user-generated text information has not been used in other video-sharing websites video classification studies.

Four types of text features — lexical, syntactic, structural, and content-specific features — have been used often in previous text-classification tasks. These four types of text features can be categorized into two broad categories: content-free features and content-specific features. Content-free features are features independent of the topics or domains of the text data and hence can be regarded as generic features. They include lexical features, syntactic features, and structural features (Zheng et al., 2006). Lexical features are used to capture lexical variations of an article in both character and word levels (Argamon, Saria, and Stein, 2003; Zheng et al., 2006) (e.g., the average word
length and the total number of characters). Syntactic features show syntactical patterns of sentences (Hirst and Feiguina, 2007; Koppel et al., 2009). These patterns can be captured by identifying function words or punctuation within sentences. Structural features represent user habits of organizing an article (e.g., paragraph length and use of signature), which have been shown especially useful for online text (Abbasi and Chen, 2005b). These features can be used to identify writing styles of different authors. Content-specific features, on the other hand, are features that can be used to represent specific topics. For example, baseball videos can be easily classified into different baseball events by identifying informative content-specific keywords such as “home run,” “double play,” “strikeout,” and “hits.” Content-specific features can be either manually selected (Zheng et al., 2006) or n-gram features extracted automatically from the collection (Abbasi and Chen, 2008; Peng et al., 2003). Most of these text features can be considered for video classification on video-sharing sites based on user-generated content.

5.2.2.3 Classification Techniques

Based on our literature review, machine learning dominated the classification techniques of previous video classification studies. Among these techniques, Hidden Markov Model (HMM), Gaussian Mixture Model (GMM), and Support Vector Machine (SVM) were the most adopted ones (Guironnet et al., 2005; Lu et al., 2001; Montagnuolo and Messina, 2007; Zhou, Cao, et al., 2005).

HMM is a popular technique widely used in pattern recognition (Rabiner and Juang, 1986). The purpose of the HMM process is to construct a model that explains the occurrence of observations (symbols) in a time sequence and use it to identify other
observation sequences. Some researchers have applied HMM for video analysis and classification. Dimitrova et al. (2000) proposed to use HMM along with text and face features for video classification. Huang et al. (1999) presented four different methods for integrating audio and visual information for video classification based on HMMs. Gibert et al. (2003) used an HMM based approach to classify sport videos into the predefined genres using motion and color features. Eickeler and Muller (1999) classified TV broadcast news by using HMMs.

GMM can be used to model a large number of statistical distributions, including non-symmetrical distributions. Given feature data, a class can be modeled with a multidimensional Gaussian distribution. In image processing applications, researchers used both unsupervised (Caillol et al., 1997; Pieczynski et al., 2000) and supervised versions (Oliveira de Melo et al., 2003) of mixture models. For example, Xu and Chang (2008) adopted GMM to classify TV broadcast programs. Girgensohn and Foote (1999) used a GMM classifier to classify staff meeting videos into different shot categories (slides, audiences, and presenters).

SVM has been shown to be a powerful statistical machine learning technique (Vapnik, 1998). The basic idea of SVM is to find a linear decision boundary to separate instances of two classes within a space. While there are multiple linear decision boundaries exist in the space, SVM will select one with the largest margin, which is the total distance between a decision boundary and the closest instances of each class. Ideally, larger margin suggest a lower classification error while new instances added into the space. SVM has two characteristics which make it efficient for classification tasks.
First, the prior knowledge is not required for it to obtain a high generalization performance and it can perform consistently with very high input dimensions. Second, SVM can obtain a global optimal solution and is especially suitable for solving classification problems with small samples (Ma and Zhang, 2003). In addition, SVM has shown excellent video classification performances (Jing et al., 2004; Lazebnik et al., 2006; Zhang et al., 2007). For example, Zhou et al. (2005) used SVM to classify soccer videos into different scenes (long shot, medium shot, or others) and reported over 92% average precision. Lin and Hauptmann (2002) applied SVM-based multimodal classifiers and probability-based strategies to continuous broadcast videos, and classified them into news and weather report categories. The results showed that the precision of SVM-based multimodal classifiers was up to 1 and significantly better than probability-based strategies.

5.3 Research Gaps and Research Questions

Table 5.2 shows selected major studies in video classification, and some general conclusions can be drawn from it. For video domains, most video classification studies focused on TV program videos (D1), while few studies paid attention to the other three domains, which are movies and movie previews (D2), specific scenario videos (D3), archival videos (D4), and video-sharing website videos (D4). In terms of feature types, non-text features (i.e. low-level video features (NT-L) and mid-level/high-level video features (NT-MH)), were adopted by the majority of previous studies. Text features, video embedded text features (T-E) and user-generated text features (T-U), were rarely explored. As for classification techniques, machine learning classification techniques (T1)
Among various machine learning classification techniques, SVM, GMM, and HMM were the most used ones.

Table 5.2: Selected Major Studies of Video Classification

<table>
<thead>
<tr>
<th>Previous Studies</th>
<th>Domains</th>
<th>Feature Types</th>
<th>Techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>D1 D2 D3 D4 D5</td>
<td>NT-L NT-MH T-E T-U</td>
<td></td>
</tr>
<tr>
<td>Huang et al., 1999</td>
<td>√</td>
<td>√</td>
<td>HMM</td>
</tr>
<tr>
<td>Girgensohn and Foote, 1999</td>
<td>√</td>
<td>√</td>
<td>GMM</td>
</tr>
<tr>
<td>Zhou et al., 2000</td>
<td>√</td>
<td>√</td>
<td>Rule-based classifier</td>
</tr>
<tr>
<td>Dimitrova et al., 2000</td>
<td>√</td>
<td>√</td>
<td>HMM</td>
</tr>
<tr>
<td>Lu et al., 2001</td>
<td>√</td>
<td>√</td>
<td>HMM</td>
</tr>
<tr>
<td>Pan and Faloutsos, 2002</td>
<td>√</td>
<td>√</td>
<td>Vcube</td>
</tr>
<tr>
<td>Lin and Hauptmann, 2002</td>
<td>√</td>
<td>√</td>
<td>SVM</td>
</tr>
<tr>
<td>Ma and Zhang, 2003</td>
<td>√</td>
<td>√</td>
<td>SVM, KNN</td>
</tr>
<tr>
<td>Rasheed et al., 2003</td>
<td>√</td>
<td>√</td>
<td>Mean-Shift Classification</td>
</tr>
<tr>
<td>Gibert et al., 2003</td>
<td>√</td>
<td>√</td>
<td>HMM</td>
</tr>
<tr>
<td>Duan et al., 2003</td>
<td>√</td>
<td>√</td>
<td>C-Support Vector</td>
</tr>
<tr>
<td>Xu and Li, 2003</td>
<td>√</td>
<td>√</td>
<td>GMM</td>
</tr>
<tr>
<td>Hsu and Chang, 2005</td>
<td>√</td>
<td>√</td>
<td>SVM</td>
</tr>
<tr>
<td>Luo and Boutell, 2005</td>
<td>√</td>
<td>√</td>
<td>SVM and Bayesian Network</td>
</tr>
<tr>
<td>Messina et al., 2006</td>
<td>√</td>
<td>√</td>
<td>Fuzzy C-Means</td>
</tr>
<tr>
<td>Hung et al., 2007</td>
<td>√</td>
<td>√</td>
<td>Bayesian Belief Network</td>
</tr>
<tr>
<td>Xu and Chang, 2008</td>
<td>√</td>
<td>√</td>
<td>SVM</td>
</tr>
<tr>
<td>Sharma and Elidrisi, 2008</td>
<td>√</td>
<td>√</td>
<td>M5P Trees</td>
</tr>
</tbody>
</table>

D1 = General TV program; D2 = Movie and movie preview; D3 = Specific scenario video; D4 = Archival video; D5 = Video-sharing website video; NT-L = Low-level video features (a sub-category of non-text features); NT-NH = Mid-level/high-level video features (a sub-category of non-text features); T-E = Video embedded text features (a sub-category of text features); T-U = User-generated text features (a sub-category of text features); T1 = Machine learning techniques for classification.
Based on our review of previous literature and conclusions, we have identified several important research gaps. First, with the emergence of Web 2.0, online videos from video-sharing websites (D4) surprisingly have seldom been addressed. Second, to the best of our knowledge, Sharma and Elidrisi (2008) is the only research that used user-generated information for online video classification. However, their classifier was designed for YouTube predefined categories only and the performance was not high. Third, among various user-generated text information, only video tags have been used (Sharma and Elidrisi, 2008). Geisler and S. Burns (2007) showed that the majority of YouTube tag terms can provide additional information about videos. Ding et al. (2009) also showed that YouTube taggers like to identify specific information such as date, geographical locations, scientific domains, religions, and opinion terms for videos. We believe other user-generated text information, such as video descriptions and comments, are also useful in video classification and can help address the video semantic gap problem.

To address the research gaps mentioned above, this chapter proposes a text-based video content classification framework for online video-sharing sites. The proposed framework can be used to identify videos for any topic or user interest. It aims to answer the following research questions:

- Are user-generated text features useful for online video classification?
- What user-generated text data and feature sets are most effective for online video classification?
- Which text classification technique is best for online video classification?
• Can accurate video classification results help identify cyber communities on video-sharing sites?

5.4 System Design

Figure 5.1 illustrates our proposed system design. Our design consists of three major steps: data collection, feature generation, and classification and evaluation.
5.4.1 Data Collection

The data collection process is designed to identify candidate videos for the classification task and collect associated user-generated text data. The input of our system is a set of selected keywords that represents users’ preferences and interests. The keywords are used to identify candidate videos. Various types of user-generated text information, including video titles, comments, video descriptions, etc., are then collected for those videos and stored in a database. Finally, users who generated the keywords are asked to create video categories based on their preferences, and a subset of the collection is randomly selected and manually classified into those categories by the users. The classification results are split into training dataset and testing dataset which will be used later for building and evaluating classifiers respectively.

5.4.2 Feature Generation

The feature generation process aims to generate text features from the collected text data that can best represent candidate videos. Three types of text features, i.e., lexical features, syntactic features, and content-specific features, are adopted in our system and denoted as F1, F2, and F3 respectively. These features have been considered in various text classification research (Abbasi and Chen, 2005b; Zheng et al., 2006; Abbasi and Chen, 2008; Abbasi, Chen, and Nunamaker, 2008; Abbasi, Chen, and Salem, 2008), but rarely in video classification studies. Several feature sets are constructed by combining different feature types and applying feature selection techniques.

5.4.2.1 Feature Extraction

In this study, we examined three features: lexical features, syntactic features, and
content-specific features. Structure features are not considered because such features (e.g., font size, font color, greetings, etc.) are not presented in video text.

Lexical features consist of character-based and word-based features (Zheng et al., 2006) and have been widely used in previous authorship research. For example, de Vel (2000), Forsyth and Holmes (1996), and Ledger and Merriam (1994) utilized different character-based lexical features in their studies. Some word-length frequency features were used in Mendenhall (1887) and de Vel (2000). In this study, we adopted 82 lexical features, including both character-based and word-based lexical features used in previous studies.

As suggested by Zheng et al. (2006), syntactic features, which include function words and punctuation words, are often used to identify styles of articles in the sentence level. Several sets of function words have been proposed in previous research (Baayen et al., 1996; Tweedie and Baayen, 1998). We adopted the set of 149 function words used in Zheng et al. (2006) because of its coverage. In addition, 8 punctuation words suggested by Baayen et al. (1996) were also used in our syntactic feature set.

Content-specific features are relevant to specific application domains and are important for online video classification. In this study, we adopted word-, character-, and POS tag unigrams, bigrams, and trigrams. In addition to n-gram features, specific user-provided video tags and video categories were also included as binary features.

The complete feature list used in our study is shown in Table 5.3. We believe our study is one of the few in examining comprehensive text features for video classification on video-sharing sites.
Table 5.3: Text Features Adopted

<table>
<thead>
<tr>
<th>Features</th>
<th>Descriptions</th>
<th>Feature Counts</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Lexical features (F1)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Character-based features</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Total number of characters</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>2. Total number of alphabetic characters</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>3. Total number of upper-case characters</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>4. Total number of digit characters</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>5. Total number of white-space characters</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>6. Total number of tab spaces</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>7-32. Frequency of letters</td>
<td></td>
<td>26</td>
</tr>
<tr>
<td>32-53. Frequency of special characters</td>
<td></td>
<td>21</td>
</tr>
<tr>
<td><strong>Word-based features</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>54. Total number of words</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>55. Total number of short words</td>
<td>Words less than 4 characters</td>
<td>1</td>
</tr>
<tr>
<td>56. Total number of characters in words</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>57. Average word length</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>58. Average Sentence length in terms of word</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>59. Average Sentence length in terms of character</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>60. Total number of different words</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>61. Hapax legomena</td>
<td>Frequency of once-occurring words</td>
<td>1</td>
</tr>
<tr>
<td>62. Hapax dislegomena</td>
<td>Frequency of twice-occurring words</td>
<td>1</td>
</tr>
<tr>
<td>63-82. Word length frequency distribution</td>
<td>Frequency of words in length of 1 to 20</td>
<td>20</td>
</tr>
<tr>
<td><strong>Syntactic features (F2)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>83-90. Frequency of punctuations</td>
<td></td>
<td>8</td>
</tr>
<tr>
<td>91-239. Frequency of function words</td>
<td></td>
<td>149</td>
</tr>
<tr>
<td><strong>Content-specific features (F3)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>POS tag n-grams</td>
<td>Unigram, bigrams and trigrams</td>
<td>Various</td>
</tr>
<tr>
<td>Character-level n-grams</td>
<td>Unigram, bigrams and trigrams</td>
<td>Various</td>
</tr>
<tr>
<td>Word-level n-grams</td>
<td>Unigram, bigrams and trigrams</td>
<td>Various</td>
</tr>
<tr>
<td>Video tags</td>
<td>Binary features</td>
<td>Various</td>
</tr>
<tr>
<td>Video categories</td>
<td>Binary features</td>
<td>Various</td>
</tr>
</tbody>
</table>

5.4.2.2 Feature Sets Generation

This step aims to generate feature sets by combining different types of features. These
feature sets are then evaluated in the classification and evaluation process. In this study, we adopted an incremental strategy to generate feature sets. Three feature sets were first created. The first feature set contains lexical features only (FS1). Lexical features and syntactic features are combined to generate the second feature set (FS2). The third set is constructed by combining lexical, syntactic, and content-specific features (FS3). This incremental approach has been frequently adopted in previous authorship studies (Abbasi and Chen, 2005b; Zheng et al., 2006) as it includes increasingly more complex and topic-relevant feature groups. Through this approach, we can obtain better insights into the effects of adding new feature sets to the previous ones.

For content-free features (F1 and F2), the total number of features is predefined as is shown in Table 5.3. However, for n-gram-based content-specific features (F3), the feature size varies and is usually much larger than the number of content-free features. An effective approach to reduce the number of such features is to adopt a minimum frequency threshold (Mitra et al., 1997; Jiang et al., 2004). We set the minimum frequency as 10 for n-gram-based parameters by following the setting adopted in Abbasi, Chen, and Salem (2008).

5.4.2.3 Feature Selection

Feature selection techniques have been shown to be effective in improving classification performances by removing irrelevant or redundant features in a large feature set. Duan et al. (2003) used feature selection to identify discriminating audio signals, while Borgne et al. (2007) adopted feature selection to reduce the number of image features. When dealing with hundred thousands or even more online videos
generated every day, the efficiencies of classifiers are also an important consideration. By
taking advantages from feature selection, we expect to identify a small set of features
which can not only perform as good as or even better than the whole feature set, but also
minimize the time to perform classification. In order to evaluate how feature selection
can improve the performance of online video classification, the fourth feature set (FS4)
was built by applying feature selection to FS3.

Information gain (IG) heuristic was adopted to perform feature selection. It has been
showed an efficient feature selection method that has been used in many text
categorization studies (e.g., Abbasi and Chen, 2005b; Koppel and Schler, 2003; Yang and
Pedersen, 1997). In this study, we used Shannon entropy measure (Shannon, 1948) in
which:

$$IG(C, F) = H(C) - H(C \mid F)$$

where $IG(C, F)$ is the information gain for feature $F$; $H(C) = - \sum_{i=1}^{n} p(C = i) \log_2 p(C = i)$
is the entropy across video category $C$; $H(C \mid F) = - \sum_{i=1}^{n} p(C = i \mid F) \log_2 p(C = i \mid F)$ is
the specific feature conditional entropy; $n$ is the total number of video category.

If videos are classified into two categories in data collection process and the numbers
of videos in the two categories are the same, $H(C)$ is 1. Then specific feature conditional
entropy $H(C \mid F)$ is calculated for each feature $F$. If videos contains feature $F$ are all in the
same category, $H(C \mid F)$ is 0 and $IG(C, F)$ is 1. However, if numbers of videos containing
feature $F$ of these two categories are the same, $H(C \mid F)$ is 1 and $IG(C, F)$ is 0. All features
with IG greater than 0 are selected.
5.4.3 Classification and Evaluation

To compare the performances between different classification techniques, three state-of-the-art classification techniques in text-analysis studies (e.g., Das and Chen, 2007; Zheng et al., 2006) were used to construct video classifiers: SVM, C4.5, and Naïve Bayes. SVM is a powerful statistical machine learning technique first introduced by Vapnik (1995). Due to its ability to handle millions of inputs and its good performance, SVM was used in previous authorship analysis studies (e.g. de Vel, 2000; Diederich et al., 2000). In addition, some studies has shown the excellent performances of SVM in video classification (Jing et al., 2004; Lazebnik et al., 2006). ID3 is a symbolic learning algorithm which has been extensively tested and shown its ability to compete with other machine learning techniques in predictive power (Chen et al., 1998; Dietterich et al., 1990). C4.5, an extension of ID3, is a decision-tree building algorithm developed by Quinlan (1986). Based on a divide-and-conquer strategy and the entropy measure, C4.5 focus on classifying mixed objects into categories according to attribute values of objects. Naïve Bayes classifier is a probabilistic classifier based on Bayes' theorem with strong independence assumptions, and uses the feature values of a new instance to estimate the probability of each category. It has also been used to perform text classification tasks in previous studies (Lewis, 1998; Mccallum and Nigam, 1998; Sahami, 1996). 10-fold cross-validation was used to evaluate all classifiers.

To evaluate the prediction performance, accuracy is adopted to evaluate the overall classification correctness of each classification technique (Abbasi, Chen, and Nunamaker, 2008). We use the average classification accuracy across all 10 folds as shown below.
Accuracy = \frac{\text{Number of correctly classified videos}}{\text{Total number of videos}} \quad (1)

5.5 Testbed and Hypotheses

5.5.1 Testbed

To evaluate our video classification framework, we chose YouTube as our data source. YouTube is the world’s largest video-sharing website. It provides robust APIs for searching videos and downloading user-generated text information about these videos. In this study we collected the following seven types of user-generated data for each video: descriptions, titles, author names, names of other videos uploaded by the video author (AuthorVideoName), comments, categories, and tags. The difference between tags and categories is that tags are given by authors of videos and could be any term, while categories are predefined by YouTube and selected by authors when uploading a video. We found these seven data types to be most content rich and carefully populated by the video authors.

The proposed framework can be used to identify videos for any topic or user interest. In this study, we aimed to identify extremist videos on YouTube. Many previous studies have demonstrated the need to identify illegal, extreme, or violent extremism content on the Internet (Burris et al., 2000; Schafer, 2002). Chen et al. (2008) showed that Web 2.0 has become an effective grassroots communication platform for extremists to promote their ideas, share resources, and communicate with each other. Extremist videos, such as suicide bombing, attacks, and other violent acts can often be found on YouTube. Therefore, automatically identifying online extremist videos has become a major research challenge for Web 2.0 (Chen et al., 2008; Salem et al., 2008).
Our testbed was created by using seventy-eight extremism-related English keywords selected by extremism study experts to search for videos on YouTube. These keywords represent major topics, ideas, and issues of interest to many domestic and international extremist groups. In total, user-generated meta data for 31,265 potentially relevant videos were collected. Those videos also included query-related videos (videos directly retrieved from YouTube using given keywords), related videos (videos related to the query-videos), and author-uploaded videos (videos uploaded by the authors of query-related videos).

To evaluate our video classification framework, 900 videos were randomly selected and tagged by extremism study experts as extremist or non-extremist videos. Among these 900 videos, 224 videos were tagged as extremist videos and 676 as non-extremist videos. In this study we included 224 extremist videos and 224 randomly selected non-extremist videos as our testbed.

5.5.2 Hypotheses

We developed the following hypotheses to examine the performances of different feature sets and classification techniques for video classification.

- H1: By progressively adding more advanced and content-rich feature sets and applying feature selection, video classification performances can be improved.
  
  - H1.1: A combination of lexical and syntactic features outperforms lexical features alone in video classification, i.e., FS2 (F1+F2) > FS1 (F1).
  
  - H1.2: A combination of content-free and content-specific (lexical and syntactic) features outperforms combination of the content-free features alone in video classification, i.e., FS3 (F1+F2+F3) > FS2 (F1+F2).
• H1.3: Applying feature selection on all feature sets can improve online video classification, i.e., FS4 (Selected F1+F2+F3) > FS3 (F1+F2+F3).

- H2: By using user-generated text data, SVM outperforms other classification techniques in video classification.
  • H2.1: SVM outperforms C4.5 in video classification by using user-generated text data, i.e., SVM > C4.5.
  • H2.2: SVM outperforms Naïve Bayes in video classification by using user-generated text data, i.e., SVM > Naïve Bayes.

5.6 Experiment Results and Discussions

For the 448 videos in our testbed, feature counts of four feature sets (FS1, FS2, FS3 and FS4) are shown in Table 5.4. The feature size was reduced from 34,229 (FS3) to 3,187 (FS4) after feature selection.

<table>
<thead>
<tr>
<th>Feature Sets</th>
<th>Feature Counts</th>
</tr>
</thead>
<tbody>
<tr>
<td>FS1 (F1)</td>
<td>574</td>
</tr>
<tr>
<td>FS2 (F1+F2)</td>
<td>1,673</td>
</tr>
<tr>
<td>FS3 (F1+F2+F3)</td>
<td>34,229</td>
</tr>
<tr>
<td>FS4 (Selected F1+F2+F3)</td>
<td>3,187</td>
</tr>
</tbody>
</table>

The experiment results of different feature types and techniques are summarized in Table 5.5 and Figure 5.2. We observed for all three classification techniques, the accuracy kept increasing as more advanced and content-rich feature types were used except using C4.5 with FS2. In addition, after applying feature selection, the accuracies
increased about 5.7% (C4.5) to 13.8% (SVM). In terms of classification techniques, SVM consistently outperformed C4.5 and Naïve Bayes with all feature sets. The best performance was achieved by using SVM with selected features of all feature types (FS4). Also, by comparing with the best performances of these three techniques, we found that among C4.5 had the worst performances. The best performance of C4.5 was only 66.09%, while it was 83.22% and 87.2% for Naïve Bayes and SVM respectively. The results indicated that C4.5 was not as efficient as the other two techniques in solving this problem. We discuss the results based on three aspects: feature types, classification techniques, and key features.

Figure 5.2: Video Classification Accuracies for Different Features and Techniques
Table 5.5: Accuracy for Different Feature Sets and Different Techniques

<table>
<thead>
<tr>
<th>Feature Set</th>
<th>C4.5 (%)</th>
<th>Naïve Bayes (%)</th>
<th>SVM (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FS1</td>
<td>59.70</td>
<td>59.21</td>
<td>61.39</td>
</tr>
<tr>
<td>FS2</td>
<td>58.05</td>
<td>61.61</td>
<td>62.51</td>
</tr>
<tr>
<td>FS3</td>
<td>61.33</td>
<td>68.80</td>
<td>73.42</td>
</tr>
<tr>
<td>FS4</td>
<td>66.09</td>
<td>83.22</td>
<td>87.2</td>
</tr>
</tbody>
</table>

5.6.1 Comparison of Feature Types

To examine the effect of adding more advanced and content-rich features and of applying feature selection, we conducted pairwise t-tests for our first hypothesis group (H1). The result is showed in Table 5.6.

Table 5.6: Pairwise T-testing of Hypotheses Group 1 (H1) on Accuracy

<table>
<thead>
<tr>
<th>Hypotheses</th>
<th>C4.5 p value</th>
<th>Naïve Bayes p value</th>
<th>SVM p value</th>
</tr>
</thead>
<tbody>
<tr>
<td>H 1.1. FS2 &gt; FS1</td>
<td>0.0066**</td>
<td>&lt;0.0001**</td>
<td>&lt;0.0009**</td>
</tr>
<tr>
<td>H 1.2. FS3 &gt; FS2</td>
<td>&lt;0.0001**</td>
<td>&lt;0.0001**</td>
<td>&lt;0.0001**</td>
</tr>
<tr>
<td>H 1.3. FS4 &gt; FS3</td>
<td>&lt;0.0001**</td>
<td>&lt;0.0001**</td>
<td>&lt;0.0001**</td>
</tr>
</tbody>
</table>

Significant levels *α= 0.05 and ** α= 0.01

When using lexical features alone (FS1), the accuracies were 59.7%, 59.21%, and 61.39% for C4.5, Naïve Bayes, and SVM respectively. The result indicates that lexical features themselves are not efficient to classify videos on online video-sharing sites. One possible reason is that the user-generated text data are generally short and lexical features, vocabulary richness features, may not be useful for short text data (Tweedie and Baayen, 1998).

The t-test result of H 1.1 shows that adding syntactic features helped Naïve Bayes and
SVM significantly improve their performances, but, however, made the performance of Naïve Bayes significantly worse. In addition, the improvements of Naïve Bayes and SVM were only 0.76% and 1.68% respectively. It may also due to the short lengths of user-generated text data. Some user-generated data types, such as video title and description, often contain only one or few sentences, and some other types, such as video tag and categories, consist of only terms or phases. These text data are too short to represent the users’ habits of using punctuation and function words.

Content-specific features improved the performances significantly for all classification techniques (3.3% to 10.9%), and hypothesis H1.2 was fully supported. It confirms the main assumption of this study that user-generated information does provide explicit content-specific information and can be used as efficient proxies of videos (e.g., for extremist videos, keywords such as suicide bombing and attacks appear frequently).

The experiment results also showed that feature selection can not only efficiently remove redundant or irrelevant features from large feature sets (from 34,229 features in FS3 to 3,187 features in FS4) but also significantly (t-tests of H1.3 are all supported) improve the classification performances no matter which classification technique was used.

5.6.2 Comparison of Classification Techniques

To compare the performances between different classification techniques (C4.5, Naïve Bayes, and SVM) on the accuracy of video classification for online video-sharing sites, we conducted pairwise t-test for the second hypothesis group (H2) and p values of the t-tests are shown in Table 5.7.
The testing result of H2.1 shows that SVM achieved significantly higher accuracy than C4.5 for all feature sets, which was consistent with previous studies in that SVM typically had better performances than decision tree algorithms (Diederich et al., 2000; Zheng et al., 2006). As for H 2.2, SVM also significantly outperformed Naïve Bayes for all feature sets.

Table 5.7: Pairwise T-testing of Hypotheses Group 2 (H2) on Accuracy

<table>
<thead>
<tr>
<th>Hypotheses</th>
<th>FS1</th>
<th>FS2</th>
<th>FS3</th>
<th>FS4</th>
</tr>
</thead>
<tbody>
<tr>
<td>H 2.1 SVM &gt; C4.5</td>
<td>0.0016**</td>
<td>&lt;0.0001**</td>
<td>&lt;0.0001**</td>
<td>&lt;0.0001**</td>
</tr>
<tr>
<td>H 2.2 SVM &gt; Naïve Bayes</td>
<td>&lt;0.0001**</td>
<td>0.0013**</td>
<td>&lt;0.0001**</td>
<td>&lt;0.0001**</td>
</tr>
</tbody>
</table>

Significant levels *α= 0.05 and ** α= 0.01

5.6.3 Key Feature Analysis

Since the FS4 significantly outperformed the combination of all feature types (FS3) with smaller number of features, we consider features of FS4 as key features which are likely the most significant discriminators with the least redundancy. To get more insights about key features in video classification for online video-sharing sites, we analyzed the feature distribution of FS4. Figure 5.3 shows the numbers of key features by user-generated data type. All seven user-generated data types contributed to key features. Among the 3,187 key features, 1,222 features came from names of author videos, 1,027 features came from comments, and 409 features came from video descriptions. Nevertheless, the number of key features is not sufficient to identify the importance of user-generated data types, because the numbers of original features of these data types are different. For example, from Figure 5.3, data type “Category” contributed only 40 key features and is unlikely to be an important data type. However, in Figure 5.4, which
shows the percentage of the overall features in each user-generated data type that are key features, we found 9.05% of overall features in “Category” are key features, which makes “Category” an important data type. In summary, among seven data types, "AuthorVideoName" contained the highest percentage of overall features that are key features (14.17%), and “Category” and “Tag” took second and third place respectively.

Figure 5.3 and Figure 5.4 confirmed our assumption that in addition to video tags,
which is the only user-generated text data used in previous online video classification studies (Sharma and Elidrisi, 2008), other data types can also provide informative information about the associated videos. For extremist videos on YouTube, text data created by the video authors (tag, description, etc.) and text data created by reviewers (i.e. comments) are both useful for classification.

We also conducted key feature analysis by feature types. As we can see in Figure 5.5, the content-specific features contributed most of the key features, including 2,056 features from character n-gram, 701 features from word n-gram, 125 features from POS n-gram, and 188 features from binary features. It might be due to the large size of content-specific features (32,556) comparing with those of the other two feature types (1,673 in total).

Figure 5.6 shows the percentage of the overall features in each feature type that are key features. Lexical feature had the highest usage even though it had the smallest size of feature set. Content-specific features contributed most key feature set in terms of feature numbers, but generally its feature usage percentage was lower than lexical features, which also indicates the importance of feature selection. In addition, syntactic features had the least contribution.
From Figure 5.5 and Figure 5.6, we observed that key features came from not only content-specific features (such as word-n-grams and POS tag-n-grams) but also content-free features (e.g. frequencies of function words and the number of different words), and therefore our assumption that both content-specific and content-free features (i.e. lexical and syntactic features) contributed to discriminating videos based on users’ interests were confirmed.
5.7 Evaluating the Impact of Video Classification on Social Network Analysis

Similar to blogs and forums, implicit cyber communities in online video-sharing websites can be defined by the interactions among users who have similar interests, including commenting, linking, or subscriptions (Chau and Xu, 2007; Fu et al., 2008). Video classification is very important for community detection and social network analysis in video-sharing websites because its results can be used to identify users of similar interests. Inaccurate video classification results affect not only the overall network topology of implicit cyber communities in video-sharing websites, but also individual node analyses such as centrality measures and participant roles, which are important units...
of cyber content analysis (Henri, 1992; Rourke et al., 2001).

In order to illustrate how the proposed video classification framework can improve social network analysis as compared to the keyword-based query approach, we present an example from YouTube. User-generated text data for a total of 543 videos were collected by searching the phrase “white power” that refers to white supremacy groups on YouTube. Again these videos included query-related videos, related videos, and author-uploaded videos. Relevant videos identified by our domain experts (through manually tagging), our video classification framework, and the keyword-based query approach which assumed all these videos were relevant, were used to construct the social networks respectively. Authors and reviewers of each identified relevant video were considered to have interactions and thus linked with each other. Considering the size of the generated social networks, we excluded links between pairs of YouTube users who had only one interaction. Figure 5.7 showed the social networks generated by using a spring layout algorithm, which places more central nodes near the middle.

Our video classification framework performed well in this example, with the classification accuracy as high as 76.43%. Consequently the generated social network was very similar to the actual network and revealed the overall network topology of the white supremacy group community on YouTube. For example, the actual network had 42 users and 5 connected components, while ours had 66 users and 12 components. In contrast, the keyword-based query approach generated a social network with a very different network topology due to many irrelevant videos. Its network contained as many as 379 users and 28 connected components.
Red points = users related to relevant videos; green triangles: users related to irrelevant videos. We also conducted analyses for user “barbituraSS,” who was located at the center of the actual network by calculating his or her degree and centrality measures. The results are displayed in Table 5.8.

Figure 5.7: Social networks of White Supremacy Groups on YouTube

Table 5.8: Degree and Centrality Measures of User “barbituraSS”

<table>
<thead>
<tr>
<th>Method</th>
<th>Betweenness</th>
<th>Closeness</th>
<th>Degree</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Value</td>
<td>Rank</td>
<td>Value</td>
</tr>
<tr>
<td>Actual</td>
<td>252.167</td>
<td>1</td>
<td>739</td>
</tr>
<tr>
<td>Classification</td>
<td>320.500</td>
<td>1</td>
<td>2,536</td>
</tr>
<tr>
<td>Keyword</td>
<td>320.500</td>
<td>6</td>
<td>133,057</td>
</tr>
</tbody>
</table>
Our video classification framework was also more reflective of users’ actual involvement in the community, with a more approximate measurement of betweenness, closeness, and degree ranks, compared with the keyword-based query approach. For example, both the network of our video classification framework and the actual network ranked user “barbituraSS” 1st for all the three measures mentioned above. This meant that user “barbituraSS” was the most important person in the white supremacy group community on YouTube. However, the network of keyword approach underestimated his/her importance by ranking this user 6th for betweenness and degree measures and 237th for closeness measures. This disparity is attributable to the keyword approach exaggerating the size of the community by incorporating many irrelevant videos. In sum, the results suggest that the proposed video classification framework will result in a more accurate representation of the social network structure of implicit cyber communities for online video-sharing websites and are helpful for individual node analyses, which are important for cyber content analysis.

5.8 Conclusions and Future Directions

In this chapter, we proposed a framework for text-based video content classification for online video-sharing sites. Different types of user-generated data (e.g. titles, descriptions, and comments) were used as proxies for online videos, and three types of text features (lexical, syntactic, and content-specific features) were extracted. We also adopted feature selection to improve accuracy and identify key features for online video classification. In addition, three feature-based classification techniques (C4.5, Naïve Bayes, and SVM) were compared. Experiments conducted on extremist videos on
YouTube demonstrated the good performance of our proposed framework.

Several conclusions can be drawn from our findings. First, our results show that user-generated text data is an effective resource for classification of videos on video-sharing sites. The proposed framework was able to classify online-videos based on users’ interests with accuracy rates up to 87.2%, achieved by using SVM with selected features of all feature types (FS4). Second, adding more advanced and content-rich feature sets can improve the classification performance, no matter which classification technique was adopted. Comparing with using only lexical features, using all text features increased accuracies up to 12%. Third, the feature selection process can significantly improve the classification performance. After applying feature selection, the accuracies sharply increased about 5.7% (C4.5) to 13.8% (SVM). Fourth, among SVM, Naïve Bayes and C4.5, SVM were the best classification techniques for most cases, which was consistent with the findings of previous studies (Diederich et al., 2000; Zheng et al., 2006). Finally, our case study showed that an accurate video classification method can help identify and understand cyber communities on video-sharing sites.

In the future we would like to consider both text features and non-text features for online video classification. We also intend to explore additional available user-generated data types, such as the user information of video authors and video reviewers. Moreover, we also plan to investigate other classification techniques and feature selection methods which may also be appropriate for text-based classification tasks.
CHAPTER 6: A HYBRID APPROACH TO WEB FORUM INTERACTIONAL COHERENCE ANALYSIS

6.1 Introduction

Text-based Computer-Mediated Communication (CMC) such as email, web forums, and newsgroups, and chatrooms have already become essential parts of our daily lives, providing a communication medium for various activities (Meho, 2006; Radford, 2006). Although the ubiquitous nature of CMC provides a convenient mechanism for communication, it is not without its shortcomings. The fragmented, ungrammatical, and interactionally disjointed nature of CMC discourse, attributable to the limitations of the CMC media, has rendered CMC highly incoherent (Hale, 1996).

For web discourse, coherence defines the macro-level semantic structure (Barzilay and Elhadad, 1997). Barzilay and Elhadad (1997) further pointed out that “coherence is represented in terms of coherence relations between text segments, such as elaboration, cause and explanation.” Coherence of online discourse, correspondingly, is represented in terms of the “reply-to” relations between CMC messages. The “reply-to” relationships can serve several functions, such as elaborating or complementing previous postings, greeting fellow users, answering questions, or oppugning previous messages.

Computer-Mediated Interaction (CMI) refers to the social interaction between CMC users (Walther et al., 1994). Such social interaction is built through the “reply-to” relationships between messages. Therefore, we also refer to the “reply-to” relationship as the interaction relationship between messages. A social interaction in online discourse happens if a user posts a message that has a “reply-to” relation with other users’
messages. Occasionally a user may interact with other users without specifying the messages he or she responds to. Common greeting messages like “Hi” are examples. But we can build fake “reply-to” relationships between such messages with the addressed user’s nearest message. This method does not affect the social interaction relationships between the users.

Since the “reply-to” relations between CMC messages can be used to build the social interaction between users, coherence of CMC is also called CMC interactional coherence in previous studies (E.g., Herring, 1999). However, current CMC media suffers the “disrupted turn adjacency” problem and the existed system functionalities do not contain sufficient “reply-to” information. Many researchers have pointed out the importance of automatically identifying CMC interactional coherence. Te’eni (2001) claimed that interactional coherence information is particularly important “when there are several participants” and “when there are several streams of conversation and each stream must be associated with its particular feedback.” Users of CMC systems cannot safely assume that they will receive a response to their previous message because of the lack of interactional coherence (Herring, 1999). Accurate interaction information is also important to researchers for a plethora of reasons. Interaction-related attributes help identify CMC user roles and user’s social and informational value, as well as the social network structure of online communities (Smith and Fiore, 2001; Fiore et al., 2002; Barcellini et al., 2005). Moreover, interactional coherence is invaluable for understanding knowledge flow in electronic communities and networks of practice (Osterlund and Carlile, 2005; Wasko and Faraj, 2005).
Interactional coherence analysis (ICA) attempts to accurately identify the “reply-to” relationships between CMC messages so that we can reconstruct CMC interactional coherence and present the social interaction between CMC users. Previously used ICA features include system generated attributes such as quotations and message headers as well as linguistic features such as repetition of keywords across postings (Sack, 2000; Spiegel, 2002; Yee, 2002). Previous studies suffer from several limitations. Most used a couple of specific features, whereas effective capture of interaction cues entails the use of a larger set of system and linguistic attributes (Nash, 2005). Furthermore, the techniques incorporated often ignored noise issues such as typos, misspellings, nicknames, etc., which are prevalent in CMC (Nasukawa and Nagano, 2001). In addition, there has been little emphasis on web forums, a major form of asynchronous online discourse. Web forums differ from email and synchronous forms of electronic communication in terms of the types of salient coherence cues, user behavior, and communication dynamics (Hayne et al., 2003).

In this chapter, we propose the Hybrid Interactional Coherence (HIC) algorithm for web forum interactional coherence analysis. HIC attempts to address the limitations of previous studies by utilizing a holistic feature set which is composed of both linguistic coherence attributes and CMC system features. The HIC algorithm incorporates finite state automation, where each stage captures interaction based on different feature types, for improved performance. The technique utilizes several similarity-based methods such as a sliding window algorithm and a Lexical Match Algorithm (LMA) in order to identify interaction based on message content cues irrespective of the various facets of CMC
noise (e.g., incorrect system feature usage, misspellings, typos, nickname usage). Collectively, HIC’s ability to consider a larger set of diverse coherence features while also accounting for noise elements allows an improved representation of CMC interaction.

This chapter displays how we address the data investigation task of the CSI framework. Except for the system feature matching module which is specific to Web forums, other parts of the HIC algorithm that emphasize on linguistic features and their corresponding text mining techniques can be applied in other Web 2.0 media for interaction identification.

The remainder of this chapter is organized as follows. Section 6.2 presents a review of previous interactional coherence analysis (ICA) research. Section 6.3 highlights important research gaps and questions. Section 6.4 presents a system design geared towards addressing the research questions, including the use of the HIC algorithm with an extended set of system and linguistic features. It also provides details of the various components of our HIC algorithm. Experimental results based on evaluations of the HIC algorithm in comparison with previous techniques are described in Section 6.5. Section 6.6 concludes with closing discussions and future directions.

6.2 Related Work

CMC interactional coherence is crucial for both researchers and CMC users. Interaction information can be used to identify user roles, messages’ values, as well as the social network pertaining to an online discussion. Example applications that can benefit from accurate online discourse interaction information include analyzing the
effectiveness of email-based interviewing (Meho, 2006) and chat-based virtual reference services (Radford, 2006). Interactional coherence analysis provides users and researchers a better understanding of specific online discourse patterns. Unfortunately, deriving interaction information from online discourse can be problematic, as discussed below.

6.2.1 Obstacles to CMC Interactional Coherence

Two properties of the CMC medium are often cited as obstacles to CMC interactional coherence (Herring, 1999): lack of simultaneous feedback and disrupted turn adjacency. Most CMC media are text-based so they lack audio or visual cues prevalent in other communication mediums. Furthermore, text-based messages are sent in their entirety without any overlap. These two characteristics result in a lack of simultaneous feedback. However, advanced CMC media have already provided simple solutions to address this concern. For example, newer versions of instant messaging software include audio and video capabilities in addition to the standard text functionality. These tools also show whether a user is typing a response, thereby providing response cues allowing interaction in a manner more similar to face-to-face communication. Since those solutions perform quite well, lack of simultaneous feedback is no longer a severe problem for CMC interactional coherence.

In contrast, resolving the disrupted turn adjacency problem remains an arduous yet vital endeavor. Disrupted turn adjacency refers to the fact that messages in CMC are often not adjacent to the postings they are responding to. Disrupted adjacency stems from the fact that CMC is “turn-based.” As a result, the conversational structure is fragmented, that is, a message may be separated both in time and place from the message it responds
to (Herring, 1999). Both synchronous (e.g., chatrooms, instant messaging) and asynchronous (e.g., email, forums) forms of CMC suffer from disrupted turn adjacency. Several previous studies have observed and analyzed this phenomenon. Herring and Nix (1997) found that nearly half (47%) of all turns were “off-topic” in relation to the previous turn. Recently, Nash (2005) manually analyzed data from an online chat room and found that the gap between a message and its response can be as many as 100 turns.

Figure 6.1 shows an example of disrupted adjacency taken from Paolillo (2006). The disruption is obvious in the example and is attributable to the fact that two discussions are intertwined in a single thread. The lines to the right hand side indicate the interaction relations amongst postings: two different widths are used to differentiate the parallel discussions. There is also one message that is not related to any of the other messages, posted by the user “LUCKMAN.” The middle column lists the linguistic features used in these messages, which will be introduced in Section 6.2.2.2.2.

<table>
<thead>
<tr>
<th>User</th>
<th>Feature</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ashna</td>
<td>Direct Address</td>
<td>Hi jatt</td>
</tr>
<tr>
<td>Dave-G</td>
<td>Direct Address</td>
<td>Kally I was only joking around</td>
</tr>
<tr>
<td>Jatt</td>
<td>Direct Address</td>
<td>Ashna: hello?</td>
</tr>
<tr>
<td>Kally</td>
<td>Substitution</td>
<td>I don’t think so.</td>
</tr>
<tr>
<td>Ashna</td>
<td>Direct Address &amp; Co-reference</td>
<td>How are u jatt</td>
</tr>
<tr>
<td>LUCKMAN</td>
<td>N/A</td>
<td>SSA all</td>
</tr>
<tr>
<td>Dave-G</td>
<td>Co-reference &amp; Conjunction</td>
<td>Therefore we need to talk</td>
</tr>
<tr>
<td>Jatt</td>
<td>Lexical relation &amp; Co-reference</td>
<td>Do we know each other? I’m ok how are you</td>
</tr>
</tbody>
</table>

Figure 6.1: Example of Disrupted Adjacency
The objective of ICA is to develop techniques to construct the interaction relations such as those shown in the right hand side of the example. Such message interaction relations can be further used to construct the social network structure of CMC users, leading to a better understanding of CMC and its users and providing necessary information for improving ICA accuracy. A review of previous interactional coherence analysis research is presented in the following section.

6.2.2 CMC Interactional Coherence Analysis

Common interactional coherence research characteristics include domains, features, noise issues, and techniques. Table 6.1 presents a taxonomy of these vital CMC interactional coherence analysis characteristics. Table 6.2 shows previous CMC interactional coherence studies based on the proposed taxonomy. Header information and quotations (F1 and F2) are system features, whereas features 3 to 6 (F3-F6) are linguistic features. A dashed line is used to distinguish these feature categories. The taxonomy and related studies are discussed in detail below.
Table 6.1: A Taxonomy of CMC Interactional Coherence Research

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Synchronous CMC</td>
<td>Internet Relay Chat (IRC), MUD, IM, etc.</td>
<td>D1</td>
</tr>
<tr>
<td>SMTP-based Asynchronous CMC</td>
<td>Email, Newsgroups</td>
<td>D2</td>
</tr>
<tr>
<td>HTTP-based Asynchronous CMC</td>
<td>Web Forums/BBS, Web Blogs</td>
<td>D3</td>
</tr>
<tr>
<td>Text document</td>
<td>News, articles, text files, etc.</td>
<td>D4</td>
</tr>
<tr>
<td>Feature</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Header information</td>
<td>“Reply-to” information in the header or title</td>
<td>F1</td>
</tr>
<tr>
<td>Quotation</td>
<td>Copy previous related message in one’s response</td>
<td>F2</td>
</tr>
<tr>
<td>Co-reference</td>
<td>Personal, demonstrative, comparative co-reference</td>
<td>F3</td>
</tr>
<tr>
<td>Lexical Relation</td>
<td>Repetition, synonymy, superordinate</td>
<td>F4</td>
</tr>
<tr>
<td>Direct Address</td>
<td>Mention username of respondent</td>
<td>F5</td>
</tr>
<tr>
<td>Other linguistic features</td>
<td>Substitution, ellipsis, conjunction</td>
<td>F6</td>
</tr>
<tr>
<td>Noise</td>
<td>Typo, misspellings, nicknames, modified quotations</td>
<td></td>
</tr>
<tr>
<td>Technique</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Manual</td>
<td>Manually identify the interaction</td>
<td>T1</td>
</tr>
<tr>
<td>Link-based method</td>
<td>Link messages by using CMC system features only</td>
<td>T2</td>
</tr>
<tr>
<td>Similarity-based method</td>
<td>Word match, VSM, SVM, lexical chain</td>
<td>T3</td>
</tr>
</tbody>
</table>

Table 6.2: Previous CMC Interactional Coherence Studies

<table>
<thead>
<tr>
<th>Previous Studies</th>
<th>Domains</th>
<th>Features</th>
<th>Noise</th>
<th>Techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>F1</td>
<td>F2</td>
<td>F3</td>
</tr>
<tr>
<td>Xiong et al., 1998</td>
<td>SMTP-based</td>
<td>√</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bagga et al., 1998</td>
<td>Text document</td>
<td></td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Choi, 2000</td>
<td>Text document</td>
<td></td>
<td></td>
<td>√</td>
</tr>
<tr>
<td>Smith et al., 2001</td>
<td>SMTP-based</td>
<td>√</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sack, 2001</td>
<td>SMTP-based</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Spiegel et al., 2001</td>
<td>Synchronous</td>
<td></td>
<td></td>
<td>√</td>
</tr>
<tr>
<td>Soon et al., 2001</td>
<td>Text document</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Newman, 2002</td>
<td>SMTP-based</td>
<td>√</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yee, 2002</td>
<td>SMTP-based</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Barcellinietal., 2005</td>
<td>SMTP-based</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nash, 2005</td>
<td>Synchronous</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
6.2.2.1 CMC Interactional Coherence Domains

CMC interactional coherence research has been conducted on both synchronous and asynchronous CMC since both of these modes show a high degree of disrupted turn adjacency (Herring 1999). Synchronous CMC, which includes all forms of persistent conversation, suffers from multiple, intertwined topics of conversation (Khan et al., 2002). In comparison, asynchronous CMC has a “thread” function, which is an effective method for categorizing forum postings based on a specific topic. However, the “thread” function is not perfect. Firstly, it does not show message-level interactions, which are vital for constructing the social network structure of CMC users. Instead, it is just an effort to group related messages together. Secondly, even in a single thread, subtopics might be generated during the discussion. This phenomenon, which poses severe problems for web forum information retrieval and content analysis, is called “topic decay/drift” (Herring, 1999; Smith and Fiore, 2001). Therefore, it is still necessary and important to apply interactional coherence analysis to asynchronous CMC.

Asynchronous CMC modes can be classified into two categories: SMTP-based and HTTP-based. SMTP-based modes (e.g., Usenet) use email to post messages to forums, whereas HTTP-based methods use forms embedded in the web pages. Previous research often focused on SMTP-based modes because the headers of posted messages contain what is referred to as “reply-to information” that specifically mentions the ID of the message being responded to. Loom (Donath et al., 1999), Conversation Map (Sack, 2001), and Netscan (Smith and Fiore, 2001) are all well-known tools that have been developed to show interaction networks of Usenet Newsgroups (SMTP-based). In
contrast, HTTP-based modes such as web forums and blogs do not contain such useful header information for constructing interaction networks. Consequently, there has been little work on HTTP-based CMC as illustrated by Table 6.2.

We also incorporate text documents into our taxonomy because they experience some problems similar to CMC incoherence, such as co-reference resolution (Bagga and Baldwin, 1998; Soon et al., 2001) and text segmentation (Choi, 2000). Techniques used for text document co-resolution, such as sliding windows (Hearst, 1994), lexical chains (Morris, 1988), and entity repetition (Kan et al., 1998) are applicable to all forms of text and can provide utility for CMC interactional coherence research.

6.2.2.2 CMC Interactional Coherence Research Features

Two categories of features have been used by previous CMC researchers and system developers. The first category is system features, which are functionalities provided by the CMC systems. The second one is linguistic features, which are interpersonal language cues.

Nash (2005) defined explicit features as those that “make fewer assumptions about what information is activated for the recipients.” Figure 6.2 shows features’ relative explicit/implicit properties. Features on the left side are more explicit than those on the right side. Explicit features are generally easier to use for deriving interaction patterns. In contrast, implicit features such as conjunctions and ellipsis are far more difficult to accurately incorporate for interactional coherence analysis. The various features are described in detail in section 6.2.2.2.1 below.
6.2.2.2.1 CMC System Features

CMC system features are usually only provided by asynchronous CMC systems. Header information and quotations are two kinds of CMC system features that can be used to construct interaction networks of asynchronous online discourse. Lewis and Knowles (1997) pointed out that SMTP-based asynchronous CMC systems will “automatically insert into a reply message two kinds of header information: unique message IDs of parent messages and a subject line of the parent (copied to the reply message’s subject line).” Unique message IDs of the parent message are intuitively useful for interaction identification. In contrast, subject lines of messages are less useful because different conversations in the same thread may have similar subject lines. Unfortunately, for HTTP-based modes, only the second type of header information is available. As shown in Table 6.2, most previous studies for SMTP-based asynchronous CMC systems relied on header information (F1 column) to construct interaction networks (e.g., Sack, 2001; Barcellini et al., 2005).

Quotations (F2 column), a context-preserving mechanism used in online discussions (Eklundh, 1998), are less frequently used to represent online conversations. Conversation Map (Sack, 2001) and Zest (Yee, 2002) are among the few previous studies that used

<table>
<thead>
<tr>
<th>Explicit</th>
<th>Implicit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header Information</td>
<td>Conjunction Substitution Ellipsis</td>
</tr>
<tr>
<td>Quotation</td>
<td>Lexical Relation</td>
</tr>
<tr>
<td>Direct Address</td>
<td>Co-Reference</td>
</tr>
</tbody>
</table>

Figure 6.2: Features’ Relative Explicit/Implicit Properties
automatic quotation identification to address disrupted adjacency. Barcellini et al. (2005) manually analyzed quotations and used them to identify participants’ conversation roles.

Although header information and quotations are effective for identifying interaction and should result in high precision intuitively, in reality they suffer several drawbacks. From the systems’ point of view, only asynchronous CMC systems contain such features. Moreover, header information provided by HTTP-based asynchronous CMC systems is of little value in many cases where the subject lines of all subsequent messages are similar or even identical. Furthermore, from the users’ point of view, some participants do not use system features and others may not use system functions correctly (Lewis and Knowles, 1997; Eklundh and Rodriguez, 2004). For instance, interaction cues may appear in the message body. Finally, some messages can interact with multiple previous messages and system features may not be able to capture such multiple interactions. As a result, using system features alone fails to consider such idiosyncratic user behavior, resulting in an incomplete representation of CMC interaction.

As is shown in Table 6.2, previous research on SMTP-based asynchronous CMC relied mostly on system features to construct the interaction network. CMC systems incorporating system and linguistic features for identification of interaction patterns, such as the Conversation Map system proposed by Sack (2000), are a rarity. The Conversation Map system also constructs interaction networks primarily using system features, but then uses the message content to construct semantic networks, which display the discussion themes for interacting messages (Sack, 2000).
The content of messages, which can be represented by various linguistic features, may be useful to complement system features in constructing CMC interactions and in many cases may be even more important (Nash, 2005). Therefore, our approach utilizes both CMC system and linguistic features to construct the interaction network with the intention of creating a more accurate representation of CMC interactional coherence and its social network structure. Important linguistic features are discussed in the following section.

6.2.2.2.2 Linguistic Features

Linguistic features are interpersonal language cues and content-based features. Previous research on synchronous CMC systems had to rely on linguistic features to construct interaction networks, since no system features were available. Several linguistic features for online communication have been identified by previous research. Three prevalent features are direct address, lexical relations, and co-reference (Halliday and Hasan, 1976; Herring, 1999; Spiegel, 2001; Nash, 2005).

Direct address takes place when a user mentions the username of another user whom he or she is addressing in the message. Coterie (Spiegel et al., 2001), a visualization tool for conversation within Internet Relay Chat, looks for direct addresses of specific people to construct the interaction network. It is important to note that addressing someone is different from referencing someone. Take the following sentence as an example: “John, take care of your brother Tom.” The speaker is addressing (and interacting) with “John” only, although “Tom” is also referenced.
Lexical relations occur when a lexical item refers to another lexical item by having common meanings or word stems. Its most common forms are repetition and synonymy (Nash, 2005). Lexical relations have also been widely used in previous studies of synchronous CMC systems. For example, Choi et al. (2000) used repetition of keywords to identify relationships between messages. Techniques that compare text similarities are often used for identifying lexical relations, where two messages are considered to have an interaction if their similarity is above some pre-defined threshold (Bagga and Baldwin, 1998).

Co-reference also occurs when a lexical item refers to another one; however such a relationship can only be identified by the context instead of the word meanings or stems. Personal co-reference is most commonly used in CMC. For example, the word “you” is frequently used to refer to the person a message addresses. Other co-references include demonstrative co-reference, which is made on the basis of proximity, and comparative co-reference, which uses words such as “same,” “similar,” and “different” (Nash 2005).

Some other linguistic features identified by previous studies include: conjunctions (e.g. but, however, therefore), substitution (e.g. “I think so.”), ellipsis (e.g. “Guess that would not be easy.”), etc. (Nash, 2005). These features have rarely been incorporates in previous studies due to the difficulty in identifying such features and their lack of prevalence in online discourse. Figure 6.1 shows an example that includes most linguistic features mentioned here.

Looking back to Table 6.2, we can see that most previous studies only utilized one or two specific features. Only Nash (2005) manually identified multiple linguistic features
for an online chatroom and found three of them to be dominant. Lexical relations covered 51% of the interaction pattern, whereas direct address and co-reference covered 28% and 15%, respectively.

6.2.2.3 Noise Issues in ICA

In ICA, noise can be defined as obstacles to direct or exact match of various features. Noise can have a profound impact on the performance of automated approaches for identifying interaction patterns. It is highly prevalent in free text, diminishing feature extraction capabilities for text mining (Nasukawa and Nagano, 2001). Typos and misspellings are common types of noise for online discourse and they exist in both direct address and lexical relations. There are also some specific forms of noise for various features, which are discussed below.

In direct address, Nash (2005) pointed out that many CMC users use nicknames to address each other (e.g., “Martin” for user “MartinHilpert,” “binary” for user “binarymike”). In addition, some usernames or their nicknames are common words; hence we need to differentiate common usage of such words with their usage as a username. For example, the word “endless” can be used to mention user “EndlessEurope.” However, “endless” might also be a common adjective in some messages. Consequently, simply comparing each word with all the usernames will not identify all the direct addresses.

In lexical relations, repetition of keywords has been used in previous research (Choi et al., 2000; Spiegel et al., 2001); but morphological word changes often decrease its performance. Word stem repetition, an improved method, can be used to solve this
problem (Reynar, 1994; Ponte and Croft, 1997). However it still cannot alleviate the effect of typos and misspellings.

Even in quotations, which are generated by the system automatically, noise still exists. Newman (2002) noticed that sometimes there were differences between the line partitions in original messages as compared to the quoted versions. Moreover, users often engage in “partial quotation” where a specific portion or segment of the original message is quoted in the reply (Eklundh, 1998).

As is shown in Table 6.2, Newman’s study (2002) is one of the few which addressed noise-related issues. He matched quotations based on sentences or sentence parts instead of matching them as a whole in order to compensate for partial quotation. In contrast, other studies failed to compensate for the existence of noise in CMC postings.

6.2.2.4 CMC Interactional Coherence Analysis Techniques

In light of the fact that several types of features can be used for interactional coherence analysis, many different techniques have previously been used to construct interaction patterns. These can be classified into three major categories: manual analysis, link-based techniques, and similarity-based techniques.

Eklundh and Rodriguez (2004) manually identified lexical relations, direct address, and co-reference for one specific online discussion. Similarly, Nash (2005) identified and extracted six linguistic features for an English chatroom. Barcellini et al. (2005) manually analyzed quotations and used them to identify participants’ conversation roles. Manual analysis of CMC interactional coherence has the obvious advantage of accuracy.
However, its disadvantage is also obvious: it is difficult to apply to large date sets and is labor intensive.

Link-based techniques construct interaction patterns using system features or rules based on message sequences. These techniques are highly prevalent in previous research because of their representational simplicity as compared to techniques that focus on linguistic features. Direct linkage techniques link messages based on header information and quotations. For residual messages unidentified by direct linkage, naïve linkage (Comer and Peterson, 1986) has been used. Naïve linkage is a rule-based technique which proposes that a message is related to all prior messages in the same discussion or the first message in the same discussion. The advantage of link-based techniques is that they are easy to implement. However link-based techniques depend on the assumption that CMC users utilize system features correctly. Moreover, naïve linkage is of low accuracy and often over-generalizes participation patterns due to its simplistic rule-based properties.

Similarity-based techniques typically use content similarity to construct interaction patterns. These techniques focus on uncovering interaction cues found in the message texts to provide insight into interactional coherence. The simplest method is exact match or direct match, which tries to identify repetition of words, word phrases, or even sentences (Choi et al., 2000; Spiegel et al., 2001). More advanced similarity-based techniques include Vector Space Model, which has been used for the cross-document coreference solution (Bagga and Baldwin, 1998) as well as to identify quoted messages (Lewis and Knowles, 1997), and lexical chains, which are often created using WordNet.
for text summarization and interaction identification (Barzilay et al. 1997; Sack, 2001). Similarity-based techniques are effective for identifying certain linguistic features (e.g., lexical relations and direct address). Some have been successfully applied in research related to text documents. However, similarity-based techniques are susceptible to noise and require careful selection of parameters.

6.3 Research Gaps and Questions

Based on our review of previous literature, we have identified several important research gaps. Firstly, little interactional coherence analysis has been conducted for HTTP-based asynchronous CMC. Previous research focused on USENET newsgroups and email, the headers of which contain accurate interaction information, rendering the use of system features sufficient for accurately capturing a large proportion of the interaction patterns. However, many web site and ISP forums (e.g., Yahoo, MSN) do not use the email protocol. Relying only on system features for such CMC modes can result in a significant amount of neglected interaction information. Secondly, little previous research has implemented techniques that use both CMC system features and linguistic attributes for interactional coherence analysis. The use of a more holistic feature set comprised of features occurring in messages headers and bodies could greatly improve interaction recall. Finally, there has been little emphasis in previous research that takes into account the impact of noise in CMC interaction networks.

Based on the research gaps identified, we propose the following research questions:

1) How effectively can we analyze interactional coherence for HTTP-based web forums using automated techniques?
2) How can techniques that use both CMC system and linguistic features improve interaction representational accuracy as compared to methods that only utilize a single feature category?

3) What impact do forum dynamics (i.e., user system usage behavior) exert on interaction representational accuracy?

4) How does noise affect the accuracy of automatically constructed CMC interaction networks?

6.4 System Design: Hybrid Interactional Coherence System

In order to address these research questions, we developed the Hybrid Interactional Coherence (HIC) algorithm to perform more accurate interactional coherence analysis, that is, to identify the “reply-to” relationships between CMC messages. The algorithm has three major components: system feature match, linguistic feature match, and residual match. System feature match and the direct address part of the linguistic feature matching component are used to identify interactions stemming from relatively more explicit features (such as headers, quotations, and direct addresses). The lexical relation match and rule-based module (which derive interaction patterns from relatively implicit cues), are only utilized when more explicit features are not present in a posting.

Several major types of noise have also been addressed.

System features used in our implementation include both headers and quotations. With header information, unique IDs of parent messages are checked first. Message subject lines are also analyzed and used. With quotations, our algorithm can identify not only normal quotations but also two special types of quotation, that is, multiple
quotations and nested quotation (Barcellini et al., 2005). The algorithm overcomes quotation noise by using a sliding window method, which compares part of the quotation to previous messages. The sliding window method has been successfully used in text similarity detection and authorship discrimination (Nahnsen et al., 2005; Abbasi and Chen, 2006). Compared with the sentence-level matching approach adopted by Newman (2002), the sliding window is better at dealing with quotation modifications made by systems or user because it is a character-level method (i.e., it compares substrings).

With respect to linguistic features, our algorithm mainly uses direct address and lexical relations. For direct address, besides traditional simple name match, our algorithm uses Dice’s equation to overcome noise such as typos, misspellings, and nicknames. Dice’s equation uses character-level n-gram matching to identify semantically related pairs of words (Adamson and Boreham, 1974). We also differentiate common words and usernames by using a lexical database and automatically generated part-of-speech (POS) tags. For lexical relations, a Lexical Match Algorithm (LMA), developed based on the Vector Space Model, which is frequently used in information retrieval (Salton and McGill, 1986), is adopted.

A comprehensive residual matching mechanism is developed for the remaining messages. It improves the naïve linkage method (Comer and Peterson, 1986) by matching messages based on their context and co-reference features. Figure 6.3 shows our system design. Details of each component are presented below.
6.4.1 Data Preparation

The data preparation component is designed to extract messages and their associated meta data from web forums. All relevant header information is extracted first. Then each message’s quotation part and body text are separated using a parser program. The parser program was also designed to deal with two special types of quotation, nested quotation and multiple quotations. Nested quotation happens when a message which already contains quotations is quoted. The parser program only parses the quotation that is nearest to the message. Sometimes users respond to different quotations in one message,
which is referred to as “multiple quotation.” The parser program parses all the related quotations. The final step of data preparation is to extract other relevant information from each message, such as author screen names, date stamps, message subjects, etc.

6.4.2 HIC Algorithm: System Feature Match

6.4.2.1 Header Information Match

In header information match, unique message IDs of parent messages, if available, are used to identify interaction. Subject lines of messages in the same thread are often consistently similar with each other if they are automatically generated by CMC systems. However, if CMC users intentionally embed interaction cues within them, subject lines can be used to identify interaction patterns as well.

6.4.2.2 Quotation Match

In quotation match, the quotation part of each message is compared with the body text of previous messages. As previously mentioned, CMC systems may modify the format of quotations (Newman, 2002), whereas CMC users may modify quotations to save space (Eklundh, 1998). Therefore, in our system the quotation part of each message is first searched for in the body text of all previous messages, referred to as “simple match.” If simple match fails due to the various aforementioned forms of noise, a sliding window method is triggered.

A sliding window method breaks up a text into overlapping windows (substrings) and compares each window against previous body texts (Kjell et al., 1994; Nahnsen et al., 2005; Abbasi and Chen, 2006). The system assigns the message (i.e., creates an interaction link) to the quoted message with the highest number of matching windows.
The following example shows how a sliding window method with a window size of 10 characters and a jump interval of 2 characters can be used to identify modified quotations.

<table>
<thead>
<tr>
<th>Original Message</th>
<th>Quoted Content</th>
<th>Message Text Windows</th>
<th>Quoted Text Windows</th>
</tr>
</thead>
<tbody>
<tr>
<td>“What do you prefer?”</td>
<td>“…do you prefer?”</td>
<td>“What do yo” “at do you” “ do you pr?” “o you pref” “you prefer”</td>
<td>“…do you” “do you pr” “o you pref” “you prefer”</td>
</tr>
</tbody>
</table>

6.4.3 HIC Algorithm: Linguistic Feature Match

Linguistic features are used to complement system features in constructing CMC interaction patterns. Nash (2005) found that direct address, lexical relations, and co-reference were three dominant linguistic features. Therefore, our hybrid interactional coherence algorithm mainly uses direct address and lexical relations in linguistic feature match, whereas the co-reference feature is indirectly used in residual match.

6.4.3.1 Direct Address Match

In direct address match, each word of a message is compared to the screen names of previous messages’ authors. By only considering authors that have appeared in prior postings within the same thread, we reduce the possibility of incorrectly considering username references to be direct addresses. For the previous example “John, take care of your brother Tom,” if user “Tom” has not already appeared in the thread, an interaction between the current message’s user and Tom will not be assigned. In situations where a direct address based interaction is found, the message containing the interaction cue is assumed to have a “reply-to” relation with the addressed users’ most recent posting.
Initially a simple match is performed in order to detect messages containing the exact same author screen names. If no simple matches are found, a Dice-based character-level n-gram matching technique is used to compensate for the effect of prevalent direct address noise in CMC such as typos, misspellings, and nicknames. The technique first uses the following Dice equation, which has been successfully used in identifying semantically related pairs of words (Adamson et al., 1974; De Roeck and Al-Fares, 2000), to estimate the similarity between a word and an author’s screen name:

\[
\text{Dice Score} = \frac{2 \times (\text{number of shared unique n–grams})}{\text{Total unique n–grams}}
\]

A pre-established experiment-based threshold is applied to improve the accuracy of direct address match. However since many CMC users choose common English words as their screen names, word sense disambiguation methods need to be applied to differentiate common usages of a word with the use of a word as a screen name. Our HIC algorithm makes use of WordNet (Miller, 1990), which has already been widely used in word sense identification (Voorhees, 1993; Resnik, 1995), to identify the meaning of words, and a POS tagger (McDonald et al., 2004) to generate the part-of-speech tags. Details of our direct address match are presented below:
Lexical Relations: The Lexical Match Algorithm

Lexical relation match assumes an interaction between the two messages that are most similar. It calculates the lexical similarities among stopword-removed messages when more explicit interactional coherence features such as quotations and direct address are not found. The key to lexical relation match is to develop an appropriate formula to calculate the similarity score. We propose a Lexical Match Algorithm (LMA) for lexical relation match. The lexical matching algorithm (LMA) is designed to identify lexical relation based interactions between postings while taking into consideration the unique characteristics of CMC interaction, such as topic drift/decay and various forms of noise (e.g., misspellings, idiosyncrasies, etc.). The algorithm measures the similarity between messages based on the content as well as turn proximity and levels of inflection and/or idiosyncratic literary variation. LMA integrates the Vector Space Model with Dice’s equation and a turn based proximity scoring mechanism.

Vector Space Model (VSM) is one of the most popular methods used to identify lexical similarities (Salton and McGill, 1986). By using word stems, VSM can also

1. For each screen name in the author list, query WordNet for meanings;
2. For each word in a message, do the following:
   2.1 Use Dice equation to find the most similar screen name appeared before;
   2.2 If the highest Dice score is greater than a predefined threshold, query WordNet for the meanings of the word and do the following:
      2.2.1 If neither the word nor the screen name has meanings, assign direct address;
      2.2.2 Else, get POS tag for the word. If the word is a noun or noun phrase, assign direct address;
      2.2.3 Else, do not assign direct address for the word.
identify morphological word changes. However, in order to identify typos, misspellings, abbreviated references, and other forms of creative user behavior, the Dice equation (Adamson et al., 1974; De Roeck and Al-Fares, 2000) is adopted in LMA to complement the traditional VSM.

Additionally, a high degree of topic decay/drift has been found in asynchronous CMC (Herring, 1999; Smith and Fiore, 2001). Nash (2005) also noticed that most CMC interactions happen within three turns. Therefore, CMC interactions represent a “closeness” characteristic, which means two closer messages are more likely to interact than two messages further away. A topic decay factor calculated by the distance (number of turns) between two messages is adopted in our LMA formula to address this “closeness” characteristic.

Here is our LMA formula for lexical similarity:

\[
\sum_{i=0}^{\text{LenX}} \sum_{j=0}^{\text{LenY}} \frac{Tf_{Xi} + Tf_{Yj}}{Df_{Xi} + Df_{Yj}} \times (\text{LenX} \times \text{LenY})^{-1} \times (\text{Distance}(X, Y) + C)^{-1}
\]

X and Y are the two compared messages. LenX and LenY are the number of unique non-stopword terms in the two messages, Xi refers to the i\textsuperscript{th} non-stopword word in message X and Yj the j\textsuperscript{th} non-stopword term in message Y. Tf is the term frequency and Df is the document frequency. Distance(X, Y) refers to the number of turns or messages between two compared messages. If there are N messages between the two compared messages, their distance is N+1. C is a constant used to control the impact of message proximity on the overall similarity between two messages.
In the formula, Dice(Xi, Yj) is used to compare two non-stopword terms. If their similarity is greater than 0.55, which is a predefined experiment-based threshold, a combined “tf-idf” score is calculated. \((\text{Len}X \times \text{Len}Y)^{-1}\) is the length normalization factor and \((\text{Distance}(X, Y) + C)^{-1}\) is the topic decay factor mentioned before. If the highest score calculated by our formula is greater than 0.002, another threshold we use, an interaction is identified. Otherwise residual match is used. The value of constant C and the two thresholds are developed based on a manually analysis of ten other threads in the LNSG forum. These 10 threads are not included in our evaluation.

6.4.4 HIC Algorithm: Residual Match

Residual match is used for messages which do not contain obvious clues for automatic interaction identification. It is utilized to help enhance interaction recall by assigning interactions based on common communication patterns. Prior residual matches have used variants of the naïve linkage method. One such implementation assigns each remaining posting (i.e., one with no identified interaction) to the first message in the thread (Comer and Peterson, 1986). Other versions of naïve linkage assign each posting to the preceding message. The intuition behind assigning each remaining post to the prior one is that messages are likely to interact with predecessors in close proximity, given the turn-based nature of CMC (Herring, 1999). Since residual matching techniques use very general assignment rules, they tend to have lower precision as compared to other techniques which use system and/or linguistic interaction cues. We propose a new rule-based residual match method which considers the message proximity as well as the
conversation structure and context. The details for our residual match are provided below:

| X: | the residual message of author A |
| Y: | previous message of author A |
| Z: | messages of other authors which are posted between Y and X and are replies to messages of author A |

1. If Y does not exist, X replies to the first message in the discussion;
2. If Y exists and Z exists, X replies to Z;
3. If Y exists and Z does not exist, X replies to what Y replies to.

The first rule is to apply the improved naïve linkage method when the residual message is the first message the author has posted in the thread. The other two rules are generated based on two human communication characteristics, which can also be found in CMC. If people give feedback or raise questions to our proposed ideas and statements, it is natural for us to comment on the feedback or answer the questions, which is characterized by the second rule. On the other hand, even if no feedback is given, people tend to strengthen or make clear their previous statements, characterized by the third rule.

6.5 Evaluation

In order to evaluate the effectiveness of our HIC algorithm, two experiments were conducted. The first experiment compared the HIC algorithm against the link and similarity-based methods. The second experiment assessed the impact of noise compensation on interaction pattern identification performance. The test bed and experimental design are described in detail below.

6.5.1 Test Bed

Our test bed consisted of two web forums. The first forum was the Sun Java Technology Forum (http://forum.java.sun.com), which is an electronic network of
practice. Analysis of such forums can help examine their social capital and knowledge contribution (Wasko and Faraj, 2005). The second one was the Libertarian National Socialist Green Party (LNSG) Forum (http://www.nazi.org/community/forum/). Analysis of such social online communities is important in order to improve our understanding of these groups and organizations (Burris et al., 2000; Schafer, 2002; Chen, 2005). Furthermore these two types of forums were selected because of their contrasting usage mechanisms and user behavior which can help evaluate the impact of forum dynamics (e.g., user system usage behavior) on interaction patterns. Users of electronic networks of practice, particularly ones pertaining to technology, are likely to be more technically savvy and less interpersonal, whereas those of social forums are more personal and closely affiliated. For both forums, several of the longest threads were studied (shown in Table 6.3).

<table>
<thead>
<tr>
<th>Forum</th>
<th>Thread No.</th>
<th>Thread Subject</th>
<th># of Messages</th>
<th># of Users</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sun Java Forum</td>
<td>1</td>
<td>Java Switch Statement</td>
<td>429</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Double precision catastrophic</td>
<td>403</td>
<td>37</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>Why use int over double?</td>
<td>453</td>
<td>36</td>
</tr>
<tr>
<td>LNSG Forum</td>
<td>4</td>
<td>Idea for banner / icon</td>
<td>148</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>Blue eyes, blond hair</td>
<td>62</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>Greetings</td>
<td>85</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>Race mixing</td>
<td>143</td>
<td>39</td>
</tr>
</tbody>
</table>

The threads in the Sun Java Technology forum were much longer than those of the LNSG forum. All seven threads were manually tagged first by a single annotator to identify their interactional coherence. A sample of one hundred messages from the
annotator was also tagged by a second coder to check the accuracy of the tagging. Both independent annotators were graduate students with strong linguistic backgrounds. The annotators determined a correct interaction by looking for interaction cues in every message. The cues included features found in message headers (e.g., an “RE:” in the subject line), quoted content from another message, linguistic cues inherent in the message body (e.g., direct address and lexical relations) as well as those based on the thread context (i.e., residual rule matching based on previous postings and interaction).

The annotators utilized the guidelines proposed by Nash (2005) for manually identifying linguistic interaction cues. Figure 6.2 provided examples of how interactional coherence could be derived using linguistic features. The inter-coder reliability across the one hundred messages had a kappa statistic of 0.88 which is considered to be reliable. The tagging results were used as our gold standard. The interaction feature breakdowns across threads based on the manual tagging are presented in Table 6.4. The difference in forum dynamics can be clearly seen. Quotations are much more prevalent in the Sun Java Technology Forum, most likely because its users are better at utilizing system functionalities. Moreover, using quotations in long threads helps readers understand the context of each message. In contrast, lexical relation is preferred in the LNSG Forum. Furthermore, the LNSG Forum members use direct address more often. This is likely attributable to the fact that people in such social groups know each other better. Finally, the high percentage of “other” features in the LNSG Forum implies that this forum’s users are more likely to display idiosyncratic and/or creative usage of CMC systems.
Table 6.4: Interaction Feature Breakdowns across Threads

<table>
<thead>
<tr>
<th>Forum</th>
<th>Thread No.</th>
<th># of Messages</th>
<th>Quotation</th>
<th>Direct Address</th>
<th>Lexical Relation</th>
<th>Others</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sun Java Forum</td>
<td>1</td>
<td>429</td>
<td>68.4%</td>
<td>14.5%</td>
<td>9.1%</td>
<td>8.0%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>403</td>
<td>70.3%</td>
<td>7.8%</td>
<td>7.6%</td>
<td>14.3%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>453</td>
<td>75.5%</td>
<td>6.4%</td>
<td>8.0%</td>
<td>10.1%</td>
</tr>
<tr>
<td></td>
<td>Overall</td>
<td>1285</td>
<td>71.5%</td>
<td>9.6%</td>
<td>8.3%</td>
<td>10.6%</td>
</tr>
<tr>
<td>LNSG Forum</td>
<td>4</td>
<td>148</td>
<td>16.2%</td>
<td>16.2%</td>
<td>41.9%</td>
<td>25.7%</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>62</td>
<td>9.7%</td>
<td>9.7%</td>
<td>53.2%</td>
<td>27.4%</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>85</td>
<td>21.2%</td>
<td>24.7%</td>
<td>35.3%</td>
<td>18.8%</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>143</td>
<td>33.6%</td>
<td>8.4%</td>
<td>33.6%</td>
<td>24.4%</td>
</tr>
<tr>
<td></td>
<td>Overall</td>
<td>438</td>
<td>21.9%</td>
<td>14.4%</td>
<td>39.5%</td>
<td>24.2%</td>
</tr>
</tbody>
</table>

6.5.2 Experiment 1: Comparison of Techniques

6.5.2.1 Experiment Setup

In the first experiment, we compared our HIC algorithm with a link-based method that relies on system features, as well as against a similarity-based method, which relies on linguistic features. These comparison techniques were incorporated since variations of the link-based method and similarity-based method have been adopted in previous studies (Spiegel et al., 2001; Soon et al., 2001; Newman, 2002; Yee, 2002). The purpose of this experiment was to study the effectiveness of the combined usage of system features and linguistic features, as done in the proposed HIC algorithm, over techniques mostly utilizing a single category of features.

The link-based method uses the quotations in the header information for interactional coherence identification (Yee, 2002). If a quotation exactly matches previous messages, the interaction is noted between the two postings. For remaining messages, the naïve
linkage method is used, which assumes that the remaining messages are replies to the first message.

The similarity-based method consists of two parts: simple direct address match and Vector Space Model match (Bagga and Baldwin, 1998). The first part identifies interactional coherence when a word is an exact match with other authors’ screen names. The second part uses the traditional “tf-idf” score to identify lexical similarity. Threshold 0.2, shown as the best threshold by Bagga and Baldwin (1998), is used for this traditional VSM match. Precision, recall, and F-measure at both the forum and thread level were used to evaluate the performance of these methods.

\[
\text{Precision} = \frac{\text{Number of Correctly Identified Interactions}}{\text{Total Number of Identified Interactions}}
\]

\[
\text{Recall} = \frac{\text{Number of Correctly Identified Interactions}}{\text{Total Number of Interactions}}
\]

\[
\text{F-measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}
\]

6.5.2.2 Hypotheses

Given the presence of system and linguistic interaction cues in online discourse, we believe that interactional coherence identification techniques incorporating both feature types are likely to provide better performance. Therefore, we propose the following hypotheses:

H1a: The HIC algorithm will outperform the link-based method for web forum interactional coherence analysis.
H1b: The HIC algorithm will outperform the similarity-based method for web forum interactional coherence analysis.

6.5.2.3 Experimental Results

Table 6.5 shows the experimental results for all three methods. Our HIC algorithm had the best performance on both the forums in terms of precision, recall, and f-measure. The linked-based method performed better than the similarity-based method for the Sun Java Technology forum, whereas its performance was worse on the LNSG forum.

Table 6.5: Experimental Results for Experiment 1

<table>
<thead>
<tr>
<th>Forum</th>
<th>Technique</th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sun Java Forum</td>
<td>HIC Algorithm</td>
<td>0.842</td>
<td>0.878</td>
<td>0.860</td>
</tr>
<tr>
<td></td>
<td>Link-based</td>
<td>0.793</td>
<td>0.756</td>
<td>0.774</td>
</tr>
<tr>
<td></td>
<td>Similarity-based</td>
<td>0.691</td>
<td>0.719</td>
<td>0.705</td>
</tr>
<tr>
<td>LNSG Forum</td>
<td>HIC Algorithm</td>
<td>0.711</td>
<td>0.711</td>
<td>0.711</td>
</tr>
<tr>
<td></td>
<td>Link-based</td>
<td>0.560</td>
<td>0.551</td>
<td>0.555</td>
</tr>
<tr>
<td></td>
<td>Similarity-based</td>
<td>0.584</td>
<td>0.678</td>
<td>0.625</td>
</tr>
</tbody>
</table>

6.5.2.4 Hypotheses Results

Table 6.6 shows the p-values for the pair-wise t-tests conducted on the interactional coherence identification accuracies to measure the statistical significance of the results. Bolded values indicate statistically significant outcomes in line with our hypotheses. Both hypotheses, H1a and H1b, are supported.

H1a: The HIC algorithm outperformed the link-based method for both the web forums (p<0.01).

H1b: The HIC algorithm outperformed the similarity-based method for both the web forums (p<0.01).
Table 6.6: P-values for Pair-wise t-tests on Accuracy for Experiment 1

<table>
<thead>
<tr>
<th>Forum</th>
<th>Techniques</th>
<th>P-values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sun Java Forum</td>
<td>HIC vs. Link-based</td>
<td>&lt;0.001*</td>
</tr>
<tr>
<td></td>
<td>HIC vs. Similarity based</td>
<td>&lt;0.001*</td>
</tr>
<tr>
<td></td>
<td>Link-based vs Similarity-based</td>
<td>&lt;0.001*</td>
</tr>
<tr>
<td>LNSG Forum</td>
<td>HIC vs. Link-based</td>
<td>&lt;0.001*</td>
</tr>
<tr>
<td></td>
<td>HIC vs. Similarity based</td>
<td>&lt;0.001*</td>
</tr>
<tr>
<td></td>
<td>Link-based vs Similarity-based</td>
<td>&lt;0.001*</td>
</tr>
</tbody>
</table>

* P-values significant at alpha = 0.01

6.5.2.5 Results Discussion

The HIC algorithm performed better than both the link-based and similarity-based methods for our test bed. The F-measure was 8%-15% higher than the other two techniques. Such improved performance was consistent across all seven threads in our test bed, as depicted in Figure 6.4.
The enhanced accuracy of the HIC algorithm was attributable to the incorporation of both system and linguistic features and its ability to handle various forms of CMC noise. The link-based method performed better than the similarity-based method in the Sun Java Technology forum because quotation features were more prevalent in this forum as illustrated in Table 6.4. For the LNSG forum, lexical relations were more commonly used as interaction cues, resulting in the improved performance of the similarity method over the link-based method on this forum. The LNSG forum members were less likely to utilize system features, which are heavily relied upon by the link-based method.

6.5.3 Experiment 2: Impact of Noise

6.5.3.1 Experiment Setup

In the second experiment, we evaluated the effectiveness of the noise compensation components in the HIC algorithm. The HIC algorithm was compared against an implementation devoid of any noise compensation components. First, in quotation match, no sliding window was used to identify modified quotations. Second, in direct address match and lexical relation match, Dice’s equation wasn’t utilized. Thus, only simple direct address match and standard Vector Space Model for lexical relations were incorporated in the “no noise compensation” implementation. Again, precision, recall, and F-measure are used as our evaluation criteria.

6.5.3.2 Hypothesis

By not considering the noise issues, we suspect some CMC interactions cannot be detected. Since our HIC algorithm utilizes several similarity-based methods which are likely impacted by noise, we propose the following hypothesis:
H2: Addressing noise issues using our proposed HIC algorithm will improve the results of interactional coherence analysis as compared to not accounting for noise issues.

6.5.3.3 Experimental Results

Table 6.7 shows the experimental results. Our HIC algorithm has better performance on both the forums.

<table>
<thead>
<tr>
<th>Forum</th>
<th>Technique</th>
<th>Precision</th>
<th>Recall</th>
<th>F-measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sun Java Forum</td>
<td>HIC Algorithm</td>
<td>0.842</td>
<td>0.878</td>
<td>0.860</td>
</tr>
<tr>
<td></td>
<td>No Noise Compensation</td>
<td>0.798</td>
<td>0.807</td>
<td>0.802</td>
</tr>
<tr>
<td>LNSG Forum</td>
<td>HIC Algorithm</td>
<td>0.711</td>
<td>0.711</td>
<td>0.711</td>
</tr>
<tr>
<td></td>
<td>No Noise Compensation</td>
<td>0.653</td>
<td>0.640</td>
<td>0.646</td>
</tr>
</tbody>
</table>

6.5.3.4 Hypothesis Results

Table 6.8 shows the p-values for the pair-wise t-tests conducted on the interactional coherence identification accuracies of the two methods. Our hypothesis H2 is supported based on the result. Addressing noise issues using the HIC algorithm improves the results of interactional coherence analysis as compared to not accounting for noise (p<0.001).

<table>
<thead>
<tr>
<th>Forum</th>
<th>Techniques</th>
<th>P-values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sun Java Forum</td>
<td>HIC vs. No Noise Compensation</td>
<td>&lt;0.001*</td>
</tr>
<tr>
<td>LNSG Forum</td>
<td>HIC vs. No Noise Compensation</td>
<td>&lt;0.001*</td>
</tr>
</tbody>
</table>

* P-values significant at alpha = 0.01
6.5.3.5 Results Discussion

The HIC algorithm’s F-measure was around 6% higher than that of the implementation with no noise compensation. Figure 6.5 shows the F-measure performance of the two methods for the seven threads. The HIC algorithm outperformed HIC with no noise compensation in all seven threads. Noise had a slightly larger effect on the LNSG forum than on the Sun Java Forum. A possible explanation is that users of technology forums compose messages more carefully than users in social forums. The Sun Java forum members are computer programmers with greater technical prowess, while the LNSG forum members are more creative in terms of their usage of language and electronic communication media. The experimental results demonstrate the impact of noise on CMC interaction networks as well as the effectiveness of noise compensation components in the HIC algorithm.

![Figure 6.5: Experiment 2 F-measure Performance for Each Thread](image-url)
6.5.4 Evaluating the Impact of Interaction Representation: An Example

Interaction networks can be used to generate the social network structure of CMC users. Inaccurate or incomplete interaction patterns have an obvious impact on overall network topology, and also on individual node metrics (e.g., degree and centrality). Such incorrect individual node statistics can affect participant role and interaction measures, which are important units of CMC content analysis (Henri, 1992; Rourke et al., 2001).

In order to illustrate how the HIC algorithm can improve social network analysis metrics as compared to previous techniques, we present an example from the Java forum. A user called “krebsnet” from the Sun Java forum that initiated thread #1 of our test bed is analyzed. The user’s degree and centrality measures generated by the various methods are shown below, in comparison with the values generated based on the manual interaction tagging (which is once again deemed the gold standard).

<table>
<thead>
<tr>
<th>Technique</th>
<th>Centrality</th>
<th>Degree</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Betweenness</td>
<td>Closeness</td>
</tr>
<tr>
<td>Actual (Manual)</td>
<td>97.072</td>
<td>80.00</td>
</tr>
<tr>
<td>HIC Algorithm</td>
<td>139.079</td>
<td>79.00</td>
</tr>
<tr>
<td>Linkage</td>
<td>206.377</td>
<td>68.00</td>
</tr>
<tr>
<td>Similarity Match</td>
<td>212.969</td>
<td>64.00</td>
</tr>
</tbody>
</table>

As shown in Table 6.9, our HIC algorithm is most reflective of the user’s actual involvement in the thread, with a more approximate measurement of centrality and degree. The other techniques exaggerate the user’s degree and centrality, which is shown in Figure 6.6. Based on the thread-level interaction results from the three methods above,
the networks shown in Figure 6.6 were generated using a spring layout algorithm, which places more central nodes near the middle. The circled point represents the user “krebsnet.” Figure 6.6 shows that the linkage and similarity match methods tend to over-assign messages to this initial poster. This is evident based on the spatial location and number of links for “krebsnet” in the linkage and similarity match methods. The social networks generated using the prior methods have a percentage error of over 100% for the betweenness and degree measures for the example node provided. The comparison techniques are off by as much as 180% regarding the node’s degree measure. In addition to differences in the absolute metric values, the degree and centrality ranking for the user (relative to other posters in the thread) is also greatly exaggerated by the link and similarity based methods. Both these comparison techniques rank “krebsnet” first in terms of degree, while the user is actually ranked 7th. The HIC algorithm ranks “krebsnet” sixth, closer to the poster’s actual level of importance. For the linkage method, the disparity is attributable to the naïve linkage match incorrectly assuming that residual messages are likely to refer to the initial posting. For the similarity match method, the erroneous metric values occur because the initial message/posting contains many important keywords in the thread. The similarity scores for this initial message are consequently higher when comparing it against other messages in the thread. This results in a high level of false message assignments. The results suggest that an improved thread-level interaction network will result in a more accurate representation of the social network structure of CMC users, which is important for CMC content analysis.
6.5 Conclusion

In this study we applied interactional coherence analysis to web forums. We developed a hybrid approach that uses both CMC system features and linguistic features for constructing interaction patterns from web discourse. The results show that our approach outperformed traditional link-based and similarity-based methods due to the use of a robust set of interaction features. Furthermore, the HIC algorithm also incorporates a wide array of techniques to address various types of noise found in CMC. Noise analysis results show that accounting for noise considerably improves performance as compared to methods that do not consider noise. Finally, we show that an improved representation...
of interaction networks results in a more accurate representation of the social network structure of CMC users. This is especially crucial for effective content analysis of online discourse archives.

In the future, we will work on analyzing user roles in web forums based on interaction networks generated by the HIC algorithm. We are also interested in identifying interaction across different forums so that we can understand the information dissemination patterns across multiple forums, and in exploring the effectiveness of using thread-level interaction networks to identify important threads in web forums. Another attractive direction is to apply our techniques to other CMC modes such as Blogs and Chatroom discussion. Blogs have very similar system features with web forums, including headers and quotations. Bloggers also share usage idiosyncrasies with web forum posters, such as typos and misspellings. Chatrooms, however, usually do not have system features and the chat postings are often too short to provide useful lexical information. By applying our algorithm to these two types of dataset we may be able to identify the potential differences in their interactional coherence.
CHAPTER 7: CONCLUSIONS AND FUTURE DIRECTIONS

7.1 Contributions

In Chapter 2 we developed a focused crawler for collecting Dark Web forums. A human-assisted accessibility mechanism was used to access identified forums with a success rate of over 90%. Many language-independent features, including URL tokens, anchor text, and level features, were used to allow effective collection of content in multiple languages. The crawler also used an incremental crawling approach coupled with a recall-improvement mechanism that continually re-spiders uncollected pages. Such approach outperformed the standard incremental-update strategy and traditional periodic-update method. A case study was also conducted to demonstrate the system’s utility for content analysis by providing insight into important discussion topics and interaction patterns for web forums. The chapter indicates that the proposed forum crawling system allows important entry to Dark Web forums, which facilitates better accessibility for the analysis of these online communities.

In Chapter 3, we proposed a GBS crawler that uses a graph-based tunneling mechanism and a text classifier that utilizes both topic and sentiment information. We demonstrated that sentiment information is useful for crawling tasks that involve consideration of content encompassing opinions about a particular topic. Moreover, we presented a novel graph-based method that ranks links associated with pages deemed irrelevant by utilizing labelled web graphs comprised of nodes labelled with topic and sentiment information. This method enables our crawler to learn tunneling strategies for situations where relevant pages were near irrelevant ones. Collectively, these elements
allowed GBS to outperform six comparison crawling methods including VSM, Keyword-based method, Context Graph Model, Hopfield Net, PageRank and BFS. The experimental results suggest that GBS is able to collect a large proportion of relevant content after traversing fewer pages than existing topic-driven focused crawlers. Additionally, the graph-based tunneling module utilized by GBS is computationally efficient, making it suitable for “real-time” data collection and analysis. Overall, the findings support the notion that focused crawlers that incorporate sentiment information are well suited to support Web 2.0 business and marketing intelligence gathering efforts.

Leveraging our work from the previous chapter, in Chapter 4 we aim to find other graph-based tunneling methods that can scale up to large graphs and run fast. We reviewed several types of state-of-the-art graph kernels including graph kernels based on walks and paths, graph kernels based on subtree patterns, and graph kernels based on limited-size subgraphs. Based on runtime requirements of focused crawlers and the properties of web graphs to be compared, we discussed the possibilities for those graph comparison algorithms to be applied in tunneling for focused crawlers and concluded that tree-based graph kernel is a promising candidate. We evaluated a simple subtree-based tunneling algorithm using GBS in a preliminary experiment. The algorithm only considered binary subtree with 3 nodes. The experiment results demonstrated that the proposed simple subtree methods are fast in training and scale up to large groups. Although the algorithm performed worse than random walk based tunneling algorithm proposed in Chapter 3 in F-measure, precision and recall, it displayed very similar trends in the three evaluation measures when a decay factor was applied and the performance
difference was small. Several parameters related to subtree patterns could be tuned in future studies to find out a suitable parameter settings for subtree-based graph tunneling algorithm.

In Chapter 5, we proposed a framework for text-based video content classification for online video-sharing sites. Different types of user-generated data such as video titles, descriptions, and comments were used as proxies for online videos, and lexical, syntactic, and content-specific text features were extracted. Feature selection was adopted to improve accuracy and identify key features for online video classification. In addition, feature-based classification techniques including C4.5, Naïve Bayes, and SVM were compared. Experiments conducted on extremist videos on YouTube demonstrated the good performance of our proposed framework. The results show that user-generated text data is an effective resource for classification of videos on video-sharing sites. The proposed framework was able to classify online-videos based on users’ interests with accuracy rates up to 87.2%, achieved by using SVM with selected features. Besides, the feature selection process resulted in significantly improvement on the classification performance. Our case study also suggested that an accurate video classification method can help identify and understand cyber communities on video-sharing sites.

In Chapter 6 we developed a hybrid approach that uses both CMC system features including header information and questions and linguistic features such as direct address and lexical relation for constructing interaction patterns from web discourse. The results show that our approach outperformed traditional link-based and similarity-based methods. Furthermore, the HIC algorithm also incorporates a wide array of techniques to
address various types of noise found in CMC. Noise analysis results show that accounting for noise considerably improves performance as compared to methods that do not consider noise. Finally, we show that an improved representation of interaction networks results in a more accurate representation of the social network structure of CMC users. This is especially crucial for effective content analysis of online discourse archives.

7.2 Future Directions

Although this dissertation has addressed several challenges in knowledge discovery tasks including data collection, selection, and investigation, future research will continue to broaden and deepen our understanding from the following directions:

1) Extend research to other Web 2.0 media. The dissertation has studied the three knowledge discovery tasks in popular Web 2.0 media such as forums, video-sharing sites, and blogs. However, there are still lots of new media that need to be explored, like photo-sharing sites, wikis, micro blogs, social bookmarking sites, etc. These media differ in content structure and user behaviors and necessitate new effective, efficient and scalable techniques for all the three tasks.

2) Explore other information embedded in user-generated content. Besides topic and sentiment information that have been partially addressed in this dissertation, there are other valuable information available in most user-generated content such as opinions, affects, geographical, and social information of web users. By incorporating such information, future research
will gain a better understanding of Web 2.0 data and users and facilitate the efficiency of the three tasks in various domains.

3) Deepen the analysis of user interaction information. In this dissertation we have used the identified user interaction to construct the network of cyber communities and analyze top active users. Future research will deepen the analysis by exploring temporal network analysis in order to understand the dynamic development of cyber communities and how information is diffused in such networks. It is also promising to combine graph theories with social network literature to identify effective collaborative or interaction patterns from these networks.
REFERENCES


visualization. Information Processing and Management, 30(1), 141–150.

attribution. In Proceedings of the IJCAI Workshop on Computational Approaches to
Style Analysis and Synthesis (pp. 69–72).

attribution. Journal of the American Society for Information Science and Technology,
60(1), 9–26.

Kumar, R., Raghavan, P., Rajagopalan, S., and Tomkins, A. (1999). Trawling the web for

information retrieval. In Proceedings of the Meeting of the International Federation of
Classification Societies.

Jing, F., Li, M., Zhang, H.-J., and Zhang, B. (2004). An efficient and effective region-
based image retrieval framework. IEEE Transactions on Image Processing, 13(5),
699–709.

Web pages for data extraction. In Proceedings of the 4th International Workshop on
Web Information and Data Management (pp. 69–75).

280(5360), 98-100.

matching for recognizing natural scene categories. IEEE Computer Society Conference
on Computer Vision and Pattern Recognition, 2, 2169–2178.


Proceedings of the IEEE Symposium on Information Visualization (pp. 125–130).


