ABSTRACT

TEXT MINING WITH THE EXPLOITATION OF USER'S BACKGROUND KNOWLEDGE: DISCOVERING NOVEL ASSOCIATION RULES FROM TEXT

by
Xin Chen

The goal of text mining is to find interesting and non-trivial patterns or knowledge from unstructured documents. Both objective and subjective measures have been proposed in the literature to evaluate the interestingness of discovered patterns. However, objective measures alone are insufficient because such measures do not consider knowledge and interests of the users. Subjective measures require explicit input of user expectations which is difficult or even impossible to obtain in text mining environments.

This study proposes a user-oriented text-mining framework and applies it to the problem of discovering novel association rules from documents. The developed system, uMining, consists of two major components: a *background knowledge developer* and a *novel association rules miner*. The *background knowledge developer* learns a user's background knowledge by extracting keywords from documents already known to the user (background documents) and developing a concept hierarchy to organize popular keywords. The *novel association rule miner* discovers association rules among noun phrases extracted from relevant documents (target documents) and compares the rules with the background knowledge to predict the rule novelty to the particular user (user-oriented novelty).

The user-oriented novelty measure is defined as the semantic distance between the antecedent and the consequent of a rule in the background knowledge. It consists of two components: occurrence distance and connection distance. The former considers the

co-occurrences of two keywords in the background documents: the more they co-occur, the shorter the distance. The latter considers the common connections of two keywords with others in the concept hierarchy. It is defined as the length of the shortest path connecting the two keywords in the concept hierarchy: the longer the path, the larger the distance.

The user-oriented novelty measure is evaluated from two perspectives: novelty prediction accuracy and usefulness indication power. The results show that the user-oriented novelty measure outperforms the WordNet novelty measure and the compared objective measures in term of predicting novel rules and identifying useful rules.

TEXT MINING WITH THE EXPLOITATION OF USER'S BACKGROUND KNOWLEDGE: DISCOVERING NOVEL ASSOCIATION RULES FROM TEXT

by
Xin Chen

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Computer and Information Science

Department of Information Systems

January 2006

APPROVAL PAGE

TEXT MINING WITH THE EXPLOITATION OF USER'S BACKGROUND KNOWLEDGE: DISCOVERING NOVEL ASSOCIATION RULES FROM TEXT

Xin Chen

Dr. Yi-fang Wu, Dissertation Advisor — Date
Assistant Professor of Information Systems, NJIT

Dr. Murray Turoff, Committee Member — Date
Distinguished Professor of Information Systems, NJIT

Dr. Il Im, Committee Member — Date
Assistant Professor of Information Systems, Yonsei University, Seoul, Korea

Dr. Vincent Oria, Committee Member — Date
Assistant Professor of Computer Science, NJIT

Dr. Marcia L. Zeng, Committee Member — Date
Professor of Library and Information Science, Kent State University

BIOGRAPHICAL SKETCH

Author: Xin Chen

Degree: Doctor of Philosophy

Date: January 2006

Undergraduate and Graduate Education:

- Doctor of Philosophy in Information Systems,
  New Jersey Institute of Technology, Newark, NJ, 2006

- Master of Science in Industrial Economics,
  Nanjing University, Nanjing, P. R. China, 1999

- Bachelor of Science in Biophysics,
  Nankai University, Tianjin, P. R. China, 1996

Major: Information Systems

Presentations and Publications:

Yi-fang Brook Wu and Xin Chen, "Assessing Student Learning with Text Processing Techniques," Journal of Asynchronous Learning Network, Volume 9, Issue 3, 2005.

Yi-fang Brook Wu, Quanzhi Li, Razvan Bot and Xin Chen, "Finding Nuggets in Documents: A Machine Learning Approach," Journal for American Society of Information Science and Technology, Accepted.

Xin Chen and Yi-fang Brook Wu, "Web Mining from Competitors' Websites," Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, August 2005.

Xin Chen and Yi-fang Brook Wu, "Web Mining for Business Intelligence: Discovering Novel Association Rules from Competitors' Websites," Proceedings of the 16th IRMA International Conference, San Diego, 2005.

Xin Chen and Yi-fang Brook Wu, "Knowledge Discovery from Competitors' Websites," Proceedings of the 3rd Pre-ICIS Workshop on e-Business (WeB2004), Washington, DC, 2004

Xin Chen and Yi-fang Brook Wu, "Assessing Student Learning Through Keyword Density Analysis of Online Class Messages," Proceedings of the 10th American Conference on Information Systems, New York City, 2004 (Best Paper Award).

Xin Chen and Yi-fang Brook Wu, "Automated Evaluation of Student Performance by Analyzing Online Messages," Proceedings of the 15th IRMA International Conference, New Orleans, 2004.

Yi-fang Brook Wu and Xin Chen, "eLearning Assessment through Textual Analysis of Class Discussions," Proceedings of the 5th IEEE International Conference on Advanced Learning Technologies, Taiwan, 2005

Quanzhi Li, Yi-fang Brook Wu, Razvan Bot and Xin Chen, "Automatically Finding Significant Topical Terms from Documents," Proceedings of the 11th American Conference on Information Systems, Omaha, August 2005.

Quanzhi Li, Yi-fang Brook Wu, Xin Chen and Razvan Bot, "Extracting Conceptual Terms from Medical Documents." Proceedings of the 11th American Conference on Information Systems, Omaha, August 2005.

Yi-fang Brook Wu, Bot, R., Xin Chen and Quanzhi Li, "Improving Concept Hierarchy Development for Web Returned Documents Using Automatic Classification," Proceedings of the 2005 International Conference on Internet Computing, Las Vegas, 2005.

Yi-fang Brook Wu, Quanzhi Li, Xin Chen and Razvan Bot, "Learning by Examples: Identifying Key Concepts from Text Using Pre-Defined Inputs," Proceedings of the 2005 International Conference on Artificial Intelligence, Las Vegas, 2005.

Quanzhi Li, Yi-fang Brook Wu, Razvan Bot, and Xin Chen, "Incorporating Document Keyphrases in Search Results," Proceedings of the 10th American Conference on Information Systems, New York City, 2004

Yi-fang Brook Wu, Quanzhi Li, Razvan Bot, and Xin Chen, "KIP: Keyphrase Identification Program with Learning Functions," Proceedings of the International Conference on Information Technology: Coding and Computing, Las Vegas, 2004.

Yi-fang Brook Wu and Xin Chen, "Assessing Distance Learning Students' Performance: A Natural Language Processing Approach to Analyzing Online Class Discussion Messages," Proceedings of the International Conference on Information Technology: Coding and Computing, Las Vegas, 2004.

Razvan Bot, Yi-fang Brook Wu, Xin Chen and Quanzhi Li, "A Hybrid Classifier Approach for Web Retrieved Documents Classification," Proceedings of the International Conference on Information Technology: Coding and Computing, Las Vegas, 2004.

Yi-fang Brook Wu, Latha Shanker and Xin Chen, "Finding More Useful Information Faster from Web Search Results," Proceedings of the ACM CIKM International Conference on Information and Knowledge Management, New Orleans, 2003.

Yi-fang Brook Wu and Xin Chen, "Extracting Features from Web Search Returned Hits for Hierarchical Classification," Proceedings of the 2003 International Conference on Information and Knowledge Engineering, Las Vegas, 2003.

This dissertation is dedicated to my beloved family

To my parents, parents-in-law and sister,
My beloved wife, Qing,
My son, Aidan, a source of inspiration and joy,
With whom I have shared
Many precious moments of my life

ACKNOWLEDGMENT

I would like to express my deep gratitude to Dr. Yi-fang Brook Wu, my research advisor, for her insightful guidance and enduring support throughout my dissertation research. My sincere appreciation is also given to Dr. Murray Turoff, Dr. Il Im, Dr. Vincent Oria, and Dr. Marcia Zeng, for their active participation in my dissertation committee and their valuable comments on this research. It is my great honor to have the entire committee who made professional and personal investments in my work. Without their guidance and outstanding help, the completion of this dissertation would not have been possible.

In addition, I would like to thank Razvan Bot and Quanzhi Li for the enjoyable collaborations in all research projects. The systems we together developed for the research projects greatly helped the implementation of the prototype system in my dissertation. Special thanks are given to the participants in the user study from the Information Systems department at NJIT and other universities. Thank them for taking time from their busy schedule and providing timely input.

Finally, I am grateful to all the fellow PhD students and colleagues in the Information Systems department at NJIT for their support and encouragement.

TABLE OF CONTENTS

Chapter	Page
1 INTRODUCTION	1
1.1 Problem Statement	1
1.2 Research Overview	3
1.3 Assumptions	4
1.4 Contributions	5
1.5 Dissertation Outline	5
2 LITERATURE REVIEW	7
2.1 Overview of Text Mining	7
2.1.1 What is Text Mining?	7
2.1.2 The Text Mining Research	9
2.1.3 Text Mining Process	10
2.2 Feature Extraction	13
2.3 Text Mining Techniques	16
2.3.1 Association Rules Mining	16
2.3.2 Association Rules Mining from Text	18
2.4 Interestingness Evaluation	21
2.4.1 Objective Measures	22
2.4.2 Subjective Measures	25
2.4.3 General or Domain Knowledge	29
2.5 Summary	30

TABLE OF CONTENTS
(Continued)

Chapter **Page**

3 DISCOVERING NOVEL ASSOCIATION RULES FROM TEXT 32

3.1 Overview 32

3.1.1 Definitions 33

3.1.2 The User-oriented Text Mining Framework 36

3.1.3 Deployment of the Framework 37

3.1.4 Differences from Existing Approaches 39

3.2 Discovering Novel Association Rules 40

3.2.1 Background Knowledge Development 40

3.2.2 Target Documents Selection 44

3.2.3 Feature Extraction from Target Documents 47

3.2.4 Association Rules Mining 48

3.2.5 Interestingness Evaluation 49

3.3 Summary 56

4 SYSTEM DESIGN AND IMPLEMENTATION 57

4.1 System Design 57

4.1.1 System Architecture 57

4.1.2 System Process Diagram 60

4.2 Design of Algorithms 61

4.2.1 Concept Hierarchy Developing Algorithm 61

4.2.2 Depth-first Hierarchy Traversal Algorithm 63

4.2.3 Shortest Path Searching Algorithm 64

TABLE OF CONTENTS
(Continued)

Chapter **Page**

4.2.4 Hierarchy Distance Calculating Algorithm 66

4.3 System Implementation 68

4.3.1 Main User Interface 68

4.3.2 Background Knowledge Development Window 69

4.3.3 Knowledge Discovery Window 72

4.4 Summary 74

5 USER EVALUATION 75

5.1 Research Questions 76

5.2 The Pilot Study 79

5.2.1 Pilot Study Design 79

5.2.2 Results 81

5.2.3 Correlations 83

5.2.4 Discussion 83

5.3 Formal User Study Design 84

5.3.1 Overview 84

5.3.2 Participants 85

5.3.3 Background Document Collecting 87

5.3.4 Knowledge Discovery 92

5.3.5 Rule Evaluation 95

5.4 Summary 97

6 DATA ANALYSIS AND RESULTS 98

TABLE OF CONTENTS
(Continued)

Chapter **Page**

6.1 Demographic Information of the Participants 98

6.2 System Settings and Results 99

6.2.1 Parameters and System Settings 100

6.2.2 Discovered Association Rules 103

6.3 Performance of the Novelty Measure 105

6.3.1 Format of the Raw Data 106

6.3.2 Novelty Prediction Accuracy 107

6.3.3 Novelty vs. Usefulness 112

6.4 Analysis of Affecting Factors 117

6.4.1 Number of the Number of Background Documents 117

6.4.2 User Information 120

6.5 Post-questionnaire Analysis 123

6.5.1 Closed Questions 123

6.5.2 Open-ended Questions 125

6.6 Summary 126

7 SUMMARIES AND CONCLUSIONS 128

7.1 Summary of Findings 128

7.1.1 The User-Oriented Text Mining Framework 129

7.1.2 The Background Knowledge Data Structure 129

7.1.3 The User-Oriented Novelty Measure 130

7.1.4 The Evaluation 131

TABLE OF CONTENTS
(Continued)

Chapter **Page**

7.1.5 Answers to Research Questions..... 133

7.2 Theoretical and Practical Implications 134

7.2.1 Theoretical Implications 134

7.2.2 Practical Implications..... 136

7.3 Contributions 139

7.4 Future Directions 140

7.5 Summary 142

APPENDIX A STOP WORD LIST..... 143

APPENDIX B CONSENT FORM..... 145

APPENDIX C BACKGROUND QUESTIONNAIRE..... 148

APPENDIX D EVALUATION INSTRUCTIONS..... 149

APPENDIX E RULE EVALUATION..... 150

APPENDIX F POST-EVALUATION QUESTIONNAIRE 151

APPENDIX G VC++ SOURCE CODE..... 153

REFERENCES 159

LIST OF TABLES

Table		Page
2.1	Objective Interestingness Measures for Association Rules	22
2.2	Groups of Objective Measures with Similar Properties	24
2.3	Classification of Objective Measures	24
3.1	Hierarchy Distance Calculation	54
5.1	Breakdowns of the Number of Rules	81
5.2	Some Selected Rules	81
5.3	Kappa Statistics K for Nominal Response	82
5.4	Kendall's Coefficient of Concordance W	82
5.5	Correlations	83
5.6	Summary of Participant Information	87
5.7	Summary of Background Documents	92
5.8	Summary of Target Documents	94
6.1	Demographic Information of Participants	99
6.2	Summary of the Input Parameters and Output	100
6.3	Number of Discovered Association Rules	103
6.4	Correlations between Four Measures and SN	108
6.5	Result of the Kruskal-Wallis Test	111
6.6	Wilcoxon Tests on Novelty Prediction Accuracy	111
6.7	Groups of Objective Measures	113
6.8	Correlations between All Measures and SU for Usefulness Indication	113

LIST OF TABLES
(Continued)

Table **Page**

6.9 Result of the Kruskal-Wallis Test on Usefulness Indication Power..... 116

6.10 Wilcoxon Tests on Usefulness Indication Power..... 116

6.11 Kruskal-Wallis Test on the Effect of #BGD on Novelty Prediction 119

6.12 Regression Analysis: Effects of User Factors on Novelty Prediction..... 122

6.13 Regression Analysis: Effects of User Factors on Usefulness Indication..... 122

6.14 Questions and Answers in the Post-questionnaire 124

7.1 Answers to All Research Questions..... 133

LIST OF FIGURES

Figure		Page
2.1	A diagram of text mining research	9
2.2	The detailed diagram of text mining research	10
2.3	The KDT process	11
2.4	Interestingness measures	31
3.1	An example of concept hierarchy	34
3.2	The user-oriented text mining framework	37
3.3	Keyword index	42
3.4	Background knowledge key word space	43
3.5	Occurrence distance	50
3.6	Connection distance	52
3.7	Hierarchy distance calculation	53
4.1	uMining system architecture	57
4.2	System process diagram	60
4.3	Concept hierarchy developing algorithm	62
4.4	Hierarchy depth traversal algorithm	63
4.5	Shortest path searching algorithm	64
4.6	Locating a path between X and Y through the root	65
4.7	Hierarchy distance d(X, Y) calculating algorithm	67
4.8	uMining main user interface	68
4.9	Background documents selection dialog	69

LIST OF FIGURES
(Continued)

Figure **Page**

4.10 Feature (keywords) extraction from background documents 70

4.11 Concept hierarchy development (POCA vs. WordNet) 71

4.12 Retrieve target documents from Google Scholar 72

4.13 Noun phrase extraction from target documents 73

4.14 Association rules mining and evaluation 74

5.1 User study status 85

5.2 Consent form (up) and the background questionnaire (bottom) 86

5.3 User interface of Research Assistant 88

5.4 Add an article to a category 89

5.5 Upload articles from Research Assistant 90

5.6 Online uploading an article 91

5.7 Rule evaluation interface 96

6.1 Rule distribution by novelty level 105

6.2 Format of the raw data 106

6.3 Novelty prediction accuracy (correlations with true value SN)..... 109

6.4 Usefulness indication power 115

6.5 Expected effect of #BGD on the performance of UN..... 118

6.6 Effects of #BGD on the performance of UN 119

CHAPTER 1

INTRODUCTION

The appearance of large heterogeneous document collections in electronic format and their increasing volume have increased the difficulty of finding interesting documents and understanding the documents for almost any end-user. According to the independent studies conducted by two large consulting companies, Gartner Group and Delphi Group, approximately 80 percent of all enterprise data is in the form of unstructured documents, such as e-mail, web pages, PDF files and paper contracts, in which the location of salient information within the document cannot be easily predicted. Consequently, the problem of text mining, which refers generally to the process of extracting interesting and non-trivial patterns or knowledge from unstructured documents, is attracting increased attention.

1.1 Problem Statement

Knowledge discovery tools tend to produce a huge number of patterns or rules, which makes it difficult for users to identify interesting and useful ones. In the study conducted at Stanford University, the association rules mining system generated over 20,000 rules from a subset of the census data containing about 30,000 records. Most of the rules are not useful, and "those that came out at the top, are things that were obvious" (Brin et al. 1997). Therefore, post-pruning of discovered patterns by selecting interesting ones and/or filtering out uninteresting ones, more commonly known as the interestingness problem, has become one of the major issues in data and text mining research.

Both objective measures and subjective measures have been proposed in the literature to solve the interestingness problem. Objective measures can be calculated directly from the characteristics of the patterns and the underlying data collection, without the consideration of domain knowledge and users' background. Thus, they are not sufficient in determining the interestingness of a discovered rule (Piatesky-Shapiro and Matheus, 1994). Subjective measures require the users to explicitly express their expectation and unexpectation from the data so that the system can compare the users' expressions with the discovered results. Such approaches assume that users know what they know and are able to express it in a given format, while in most text mining tasks, users do not have sufficient knowledge about the collection, nor do they know what should be expected before the results are presented.

The limitations of existing interestingness measures prevent themselves from being effectively used to evaluate the interestingness of rules discovered by a data mining system. In text mining field, even less work has been done for developing interestingness measures that can help users find interesting rules. Although user's background knowledge and interests are important, they are seldom exploited in the knowledge discovery process. Motivated by the problems described above, the study tries to develop a user-oriented text mining framework which takes into consideration users' background knowledge in the text mining process and applies such knowledge to evaluate the interestingness (novelty and usefulness) of discovered association rules among concepts. To reduce the users' effort in background knowledge development, the study also aims at developing an algorithm that can implicitly learn the users' background knowledge from

documents, instead of asking users for explicit expressions of their interests and expectations.

1.2 Research Overview

This research presents a user-oriented text-mining framework, in which a background knowledge component is introduced to the text mining process. The framework is applied to the problem of discovering novel association rules from text with the presence of the user's background knowledge derived from a set of background documents. The system, called uMining (user-oriented Text Mining), follows two steps to discover knowledge from text: background knowledge development and novel association rules mining. Background knowledge is developed from background documents, which are documents already known to the user. Keywords are extracted from background documents and organized into a hierarchical structure by using the Probability of Co-occurrence Analysis technique (POCA) (Wu, 2001). Target documents are retrieved from a large corpus by selecting documents that are relevant to the user's background. Association rules are mined among noun phrases extracted from target documents. The user-oriented novelty measure, defined as the semantic distance between the antecedent and the consequent of a rule in the background knowledge, is then calculated to predict the interestingness of discovered rules.

The performance of the user-oriented novelty measure was evaluated by comparing the results generated by the uMining system to the actual users' novelty and usefulness ratings of discovered rules. The proposed novelty measure was also compared with other interestingness measures for their performance of identifying novel and useful

rules. A user study was conducted with PhD students and the system was run to discover novel association rules from research papers for each participant. The results show that the user-oriented novelty measure has high novelty prediction accuracy, and it outperforms the WordNet novelty measure in novelty prediction. It is also found that the user-oriented novelty measure has high usefulness indication power, and it outperforms the WordNet novelty and seven compared objective interestingness measures in usefulness indication.

1.3 Assumptions

The proposed method is based upon the following assumptions. First, it is assumed that background documents are already available. The issue of how to build a collection of documents that can represent what the user knows is not addressed in the current study. Users who need to perform text mining tasks usually have a strong interest in some area, and it is reasonable to assume that they have read related documents to acquire knowledge about the topic. Even if they are at the beginning stage, information searching tools can help them collect documents of interests and develop the necessary knowledge quickly. Second, although user's knowledge is diverse and heterogeneous, in this study a user's background knowledge is restricted to a specific topic that he/she is interested in. Therefore, only documents that are related to such topic will be selected as background documents. The relevancy judgment will be made by the users when they submit their background documents. Third, the current study does not consider the dynamic changes in users' background knowledge. These assumptions will be re-examined in the future work section of this study.

1.4 Contributions

This study is the first attempt at capturing users' background knowledge and exploiting such knowledge in text mining tasks to discover personalized knowledge. The user's background knowledge is implicitly learned from a set of documents, which makes the system easy to use. The results (the rules and their novelty predictions) are customized for users with different backgrounds. The user-oriented novelty measure has the advantages of both objective measures and subjective measures.

The proposed approach could be employed in various types of knowledge discovery systems. For instances, it could be implemented as a personalized knowledge discovery system, which not only manages the users own documents, but also discover unknown knowledge from new documents. The proposed approach also provides users with new ways of searching for information and learning knowledge. For example, many people, from blog writers to scientific researchers, are writing documents to record what they have learned about something of interests or a specific research topic. Most of the time, the authors may have their own focus and are likely to neglect other important ones. By pointing out what is implicitly missing in their writings, the proposed system could help people learn new knowledge without explicitly searching for them.

1.5 Dissertation Outline

This dissertation includes seven chapters. Chapter 1 introduces the dissertation and provides an overview of this study. Chapter 2 provides literature review of the related theories and research in text mining and interestingness evaluation. Chapter 3 presents the user-oriented text mining framework, the proposed methodology, as well as the

associated algorithms. Chapter 4 describes the implementation of the system, uMining, including system architecture, system design, and important user interfaces. Chapter 5 discusses evaluation methodologies including the user study design and data collection methods. Chapter 6 starts the presentation of study results by providing descriptive statistics and quantitative data analyses. Chapter 7 concludes the dissertation and provides summaries, discussions, contributions, limitations, and future directions of the study.

CHAPTER 2

LITERATURE REVIEW

This chapter provides the review on the literature related to this study. It includes an overview of text mining research, feature extraction techniques, text mining techniques, and interestingness evaluation measures. Based upon the literature review, the main research topics are identified, and the limitations of existing approaches are discussed.

2.1 Overview of Text Mining

This overview includes the definitions of text mining, the position of text mining in a broader research diagram, and the general process of a text mining task.

2.1.1 What is Text Mining?

Text mining refers generally to the process of extracting interesting and non-trivial patterns or knowledge from unstructured text documents. It is a young interdisciplinary field which draws on information retrieval, data mining, machine learning, statistics, computational linguistic and other related fields.

Knowledge discovery is defined as "the non-trivial process of identifying valid, novel, potentially useful and ultimately understandable patterns in data" (Fayyad et al. 1996). The general purpose of knowledge discovery is to "extract implicit, previously unknown, and potentially useful information from data" (Frawley et al. 1991). Researchers in text mining field adopted the similar notion of knowledge discovery, while data here is referred to unstructured textual documents. The criteria used to

evaluate the discovered knowledge in data mining, such as validity, usefulness, and understandability, are expected in text mining as well.

A popular notion of text mining is the extension of data mining to textual data, also known as knowledge discovery in texts (KDT) (Feldman and Math, 1995). Existing data mining algorithms can be adopted after the documents are converted to structured format by information extraction (Nahm and Mooney, 2002) or document classification (Pierre, 2002) techniques. New algorithms may also be designed to solve specific text processing tasks, such as summarization, exploration of interesting patterns and trend analysis. Kodratoff (1999) made a clear comparison between text mining and natural language processing, information retrieval and information extraction. In the comparison, KDT is defined as "the science that discovers knowledge in texts, where 'knowledge' is taken with the meaning used in KDD, that is: the knowledge extracted has to be grounded in the real world and will modify the behavior of a human or mechanical agent." All the problems already described as being proper to KDD can be introduced in KDT.

Hearst (1999) defines text mining as finding nuggets from textual data, "a process of exploratory analysis that leads to the discovery of unknown information, or to answers to questions for which the answer is not currently known," such as using text to form hypotheses about disease and to uncover social impact. This definition emphasizes that discovering unknown knowledge is the key characteristics of text mining techniques.

Though it is agreed that text mining is highly related to data mining, there exist different understandings of what text mining is or should be. A consensus on the definition has not been reached.

2.1.2 The Text Mining Research

Different people look at text mining from different angles. Some address the importance of the underlying methodologies; while others focus on the techniques and the tasks to be solved. Text mining techniques rely on mathematical models and statistical methods. The types of tasks to be solved vary greatly, depending on the nature of the problem and the types of knowledge to be discovered. Some of the tasks are unique in text mining, while others are very common problems in other fields. For example, discovery of interesting concept distribution (Feldman et al. 1998) is a new task introduced in KDT, but classifying documents using frequent item set (Beil et al. 2002; Fung et al. 2003) is "an old bottle with new wine." In addition, text mining techniques have been applied in many domains, especially in Bioinformatics and business applications. The following diagram describes text mining research in general.

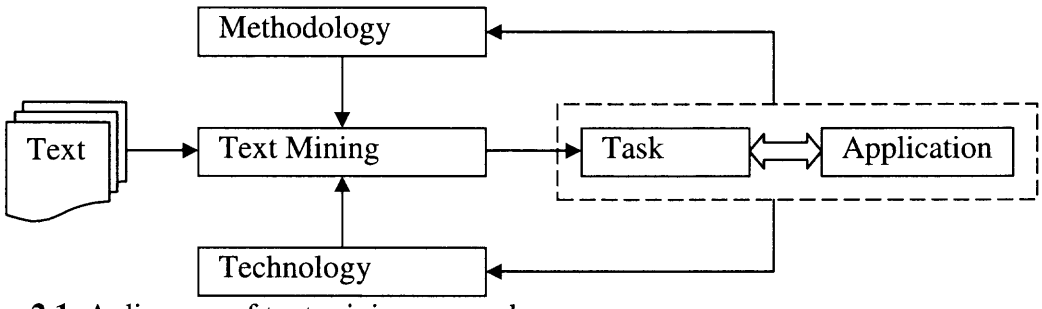
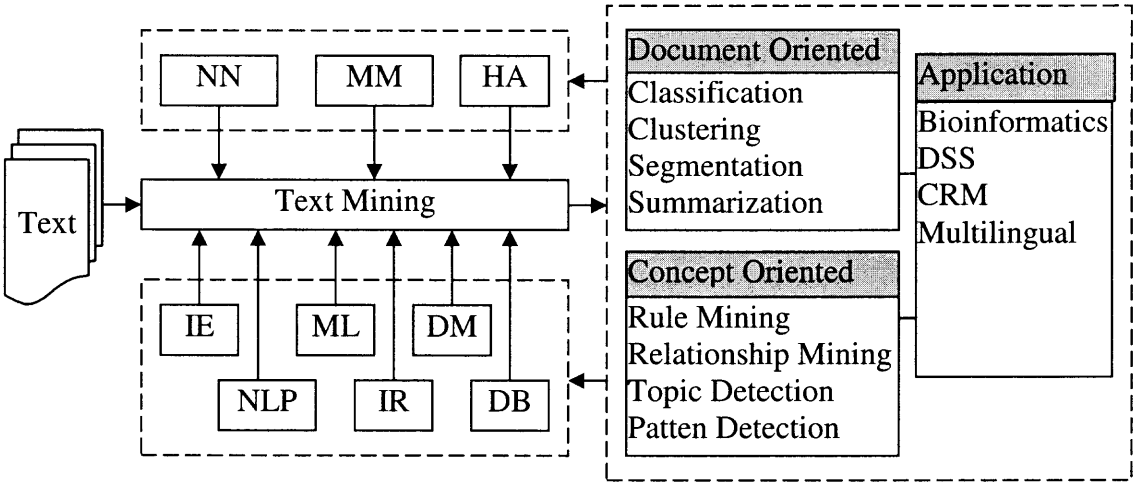


Figure 2.1 A diagram of text mining research.

As shown in Figure 2.1, under the guidance of certain methodologies and supported by technologies in related fields, text mining aims at finding interesting and useful patterns from text to fulfill certain tasks or to develop appropriate applications that solve particular problems in practice. The knowledge discovered from text will enable the system to refine the existing models, enhance the technologies in other related fields, and create new models and technologies. KDT has evolved, and continues to evolve,

from the intersection of research fields such as data mining, information retrieval, machine learning, natural language processing, pattern recognition, databases, statistics, artificial intelligence, and data visualization. The unifying goal is to extract high-level knowledge from low-level unstructured or semi-structured textual data in the context of large document sets. The detailed diagram is shown in Figure 2.2.



NN: Neural Networks
MM: Markov Model
HA: Hyperlink Analysis

IE: Information Extraction
NLP: Natural Language Processing
ML: Machine Learning
IR: Information Retrieval
DM: Data Mining
DB: Database

Figure 2.2 The detailed diagram of text mining research.

2.1.3 Text Mining Process

In general, text mining refers to the entire process of knowledge discovery from text, rather than the single stage in which the mining algorithm is performed. It consists of steps from extracting features from documents, selecting useful features, mining patterns, to presenting and utilizing the results. Though text mining generally follows the data mining process, each step of the text mining process is different from that of the data mining process. The major difference occurs in the preprocessing step, in which

particular approaches are required to extract useful features from text. The existing data mining algorithms might not be applicable to text directly, because processing the textual data is not the same as processing numeric data. For example, two strings may not be compared character by character for equality because of the variations in natural language expressions. Instead, a similarity measure may be needed to judge the equality of two strings. Additionally, the discovered patterns need to be interpreted and evaluated with different methods.

The KDD process follows the following steps: data selection, data preprocessing, data transformation, data mining, and result interpretation/evaluation (Fayyad et al. 1996). A similar process, consisting of document retrieval, feature extraction, data structure construction, text mining and evaluation, is adopted to describe the KDT process, as shown in Figure 2.3.

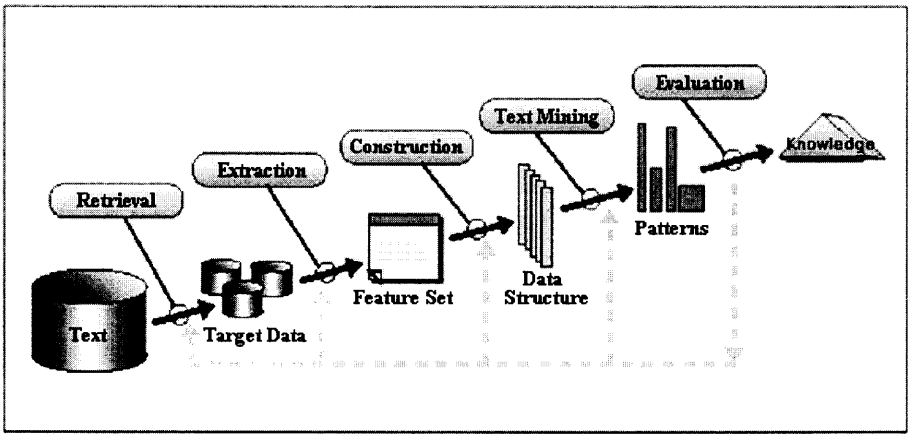


Figure 2.3 The KDT process.

The KDT process begins with generating a target data set from the raw data. The data available on hand often contains irrelevant data, which not only wastes computing resources, but also introduces noises to the result. The target data is created by gathering a set of relevant data, or selecting a subset of variables or data samples, on which

discovery is to be performed. In text mining, the target data is usually a set of documents which can be generated with information retrieval techniques.

Feature extraction is the conversion of unstructured documents to structured feature sets. This is the most important step in the preprocessing stage. Decision on what features to extract depends on the nature of the documents, the goal of the application, the expectation of extracted patterns, and so on. Examples of document features include, but not limited to, pre-defined domain vocabularies (Feldman and Math, 1995), frequent maximal word sequence (Ahonen-Myka et al. 1999; Ahonen-Myka, 2002), template-based information extraction (Nahm and Mooney, 2000a; Nahm and Mooney, 2000b; Nahm and Mooney, 2002), and category labels assigned as faceted metadata (Pierre, 2002).

Data structure construction is the development of a certain data structure from the extracted features. The data structure is not mandatory, but could allow more advanced text mining options. In the FACT system (Feldman, 1995), a manually constructed concept hierarchy, which consists of a controlled set of concepts labeling the documents, is provided to extract unexpected concept distributions. Certain data structures also allow performing mining processes at multiple levels (Han and Fu, 1999; Han and Fu, 1994).

Text mining algorithms are then performed on the structured document features and the data structure if available. With appropriate revisions or extensions, standard data mining algorithms, such as association rules mining, decision tree construction, clustering, and classification, can be applied to discover knowledge from the document features. New algorithms specifically designed for text mining tasks can also be used for knowledge discovery.

Evaluation involves judging the interestingness and usefulness of the extracted patterns. Text mining algorithms often produce a huge number of patterns, which could easily become overwhelming to users. Selecting and presenting the interesting patterns to the user is an important task in the post-processing stage.

The following review will focus on feature extraction techniques, text mining techniques and interestingness evaluation.

2.2 Feature Extraction

Feature extraction is to extract elements of interests from documents, so that the unstructured documents can be converted to structured feature sets. Mining on the features manually assigned to each document (e.g. category labels and author keywords) can be a solution, but such features are often unavailable and expensive to create. Moreover, the types of features may need to be dynamically changed to fulfill different text mining tasks. Automatic feature extraction thus becomes extremely important for text mining algorithms. Some commonly used approaches include information extraction (Feldman, 2001; Grishman, 1997; Nahm, 2000), noun phrase extraction (Brill, 1995; Church, 1988; Cutting, 1992), keyphrase extraction (Barker, 2000; Frank, 1999; Turney, 2000; Wu, 2004), and maximal sequence identification (Ahonen-Myka, 2002).

Evidences from language learning of children (Snow and Ferguson, 1997) and discourse analysis theories, e.g. Discourse Representation Theory (Kamp, 1981), show that the primary concepts in text are carried by noun phrases. Therefore, noun phrases are good document features for the purpose of mining association rules among concepts. Noun phrase extraction includes two steps: part-of-speech (POS) tagging and noun

phrase identification. POS tagging is to assign the right POS tag to each token in the text. Once each token in the free text is tagged, noun phrases can be identified by selecting the sequence of tokens whose POS tags are of interests.

There are two major approaches to POS tagging: Markov-model based and rule based approaches. Markov-model based taggers (Cutting et al. 1992; Church, 1988; Dermatas and Kokkinakis, 1995; DeRose, 1998) assign to a sentence the tag sequence that maximizes *Prob(word|tag)*Prob(tag|previous n tags)*. First, the initial POS tag of each word is assigned based solely on the probability that the word occurs with a particular tag. In other words, the tag encountered most frequently in the training set is the one assigned to an ambiguous instance of that word. It is the simplest way of tagging but has a problem with assigning a valid tag for a given word in an unreasonable sequence of tags. The second probability calculates the frequency of a given sequence of tags and tries to pick up the most feasible one. This is sometimes called the n-gram approach, referring to the fact that the best tag for a given word is determined by the probability that it occurs with the n previous tags.

Typical rule based approaches (Brill, 1992; Brill, 1995; Ngai and Yarowsky, 2000) use contextual information to assign tags to unknown or ambiguous words. These rules are often known as context frame rules. As an example, a context frame rule might say something like: if an ambiguous/unknown word X is preceded by a determiner and followed by a noun, tag it as an adjective.

In addition to contextual information, many taggers use morphological information to aid in the disambiguation process. One such rule might be: if an ambiguous/unknown word ends in an *-ing* and is preceded by a verb, label it a verb

(depending on the given theory of grammar). Some systems go beyond using contextual and morphological information by including rules pertaining to such factors as capitalization and punctuation. Information of this type is of greater or lesser value depending on the language being tagged. In German for example, information about capitalization proves extremely useful in the tagging of unknown nouns.

Rule-based taggers most commonly require supervised training, but very recently there has been a great deal of interest in automatic induction of rules. One approach to automatic rule induction is to run an untagged text through a tagger and see how it performs. A human then goes through the output of the first phase and corrects any erroneously tagged words. The properly tagged text is then submitted to the tagger, which learns correction rules by comparing the two sets of data. Several iterations of this process are sometimes necessary to obtain the best set of tags for a given text.

To overcome the disadvantages of each tagger, some researchers propose hybrid approaches and apply system combinations to identify noun phrases (Sang et al. 2000; Chen and Chen, 1994).

2.3 Text Mining Techniques

Various text mining techniques have been reported in the literature. The related work is in the area of association rules mining and its application on textual data.

2.3.1 Association Rules Mining

The goal of association rules mining (Agrawal et al. 1993; Agrawal and Srikant, 1994; Agrawal et al. 1996) is to generate all significant association rules between items in the transaction database. Such rules will help supermarket managers make decisions on promotion design, merchandise placement, and sales arrangement to increase profit.

2.3.1.1 Association Rule. The general expression of an association rule is $X \rightarrow Y$ *support/confidence*, where X and Y are sets of items or itemsets, and *support* and *confidence* are two numbers indicating the significance and the strength of the rule. X is called the antecedent and Y is called the consequent of the rule. *Support* is the probability of the joint occurrence of X and Y, and *confidence* is the conditional probability of Y given X in the database. An example of an association rule is *Bread, Butter* → *Milk 5%/90%*, which means *Bread, Butter*, and *Milk* appear together in 5% of the transactions, and 90% of the customers who purchased *Bread* and *Butter* also purchased *Milk*. Besides *support* and *confidence*, other constraints, such as syntactic constraints, may also be applied. Syntactic constraints affect which items can appear in which side(s) of the rule. For example, a syntactic constraint may specify that only those rules that have a certain item in the antecedent or the consequent should be kept in the results.

The association rule mining problem is usually decomposed into two sub-problems: frequent itemset identification and rule generation from frequent itemsets.

Frequent itemsets are combinations of items that have a greater *support* than the specified minimum support threshold. Syntactic constraints, if any, are applied to further filter out uninteresting items. For every frequent itemset, its items are partitioned into two parts: one for the antecedent and one for the consequent. Confidence is calculated according to the *support* values of the two parts and the entire itemset. If the *confidence* value is greater than the minimum confidence threshold, the rule is saved. All combinations of items in the frequent itemset have to be tested to find all validated rules. The second sub-problem is more straightforward, so the main effort is devoted to the first sub-problem: frequent itemset identification.

2.3.1.2 Algorithms. Since the introduction of the association rule mining problem, various algorithms have been proposed either to improve the efficiency of the algorithm or to apply it to a particular data set. The AIS algorithm (named after the authors' initials) was presented as an initial solution for association rule mining from large database (Agrawal et al. 1993). Candidate itemsets are generated and counted on-the-fly when the database is scanned. After reading a transaction, the algorithm examines which of the itemsets previously found to be large are contained in this transaction. New candidate itemsets are generated by expanding these large itemsets with the other items in the transaction. The AIS algorithm is straightforward, but it results in unnecessarily generating and counting too many itemsets that are actually small.

The APRIORI (Agrawal et al. 1996; Agrawal and Srikant, 1994) differs fundamentally from the AIS algorithm in terms of which candidate itemsets are counted and in the way new candidates are generated. It discovers large itemsets through multiple passes over the data. In the first pass, *support* values of individual items are counted to

determine which of them are large. The large one-item itemsets are served as the seed for the subsequent pass, during which two-item itemsets are generated and the *support* value of each of them is counted to determine if the itemset is large or not. Then, new large itemsets are appended to the seed set of itemsets. This process continues until no new large itemsets are found in a pass.

2.3.2 Association Rules Mining from Text

Association rules mining algorithms have been applied to textual data. This section reviews three approaches that discover association rules from documents.

2.3.2.1 The FACT System. The FACT system discovers associations – patterns of co-occurrence – among keywords labeling the documents in a collection (Feldman, 1998). It exploits the background knowledge of the document labels to filter the discovered rules. The knowledge discovery process is viewed as a query process, in which a discovery request is viewed as a query over the implicit set of possible results supported by a collection of documents, and where background knowledge is used to specify constraints on the desired results of this query process. Execution of a knowledge-discovery query is structured so that these background-knowledge constraints can be exploited in the search for possible results.

Background knowledge can come from many different sources, such as databases of facts about the domain and other textual information sources. User query consists of the keywords and the unary and binary predicates defined by the background knowledge. Association-discovery query has three parts. The first part specifies what types of keywords are desired in the left-hand and right-hand sides of any found associations, as well as what *support* and *confidence* the association should have. The second part of a

query (possibly empty) specifies constraints in terms of the predicates defined by the background knowledge that the user wants any found association to satisfy. The third part of a query (also possibly empty) specifies constraints on the size of the various components of the association.

The assumption is that each document is labeled with a set of keywords and the background knowledge about these keywords is available. However, in many cases, documents are not labeled, the background knowledge is unavailable, or the vocabulary in the background knowledge does not match the document labels. In addition, users have to know the background knowledge in order to formulate the query. Therefore, this technique suffers from these limitations for practical uses.

2.3.2.2 Mining from Metadata. One difficulty of applying association rule mining algorithm to text is due to the high dimensionality of documents. It is computationally expensive to use words to mine rules, and it may also generate many rules that are too obvious to be useful. Preparing the documents for association rule mining includes not only extracting features but also reducing the dimensions of the features. Mining from metadata records of documents automatically generated by document categorization technique is such a method that tries to meet the two requirements (Pierre, 2002).

A crucial part of the approach is the use of faceted metadata (English et al. 2002). Individual facets represent orthogonal conceptual dimensions. The set of facets as well as the set of possible concepts in each facet must be determined by a domain expert. The facets and the concepts in each facet are controllable, so that the metadata database schema can be customized according to the purpose of the text mining tasks. The metadata schema is not rigid and can be determined after the documents are created.

Automated text categorization techniques are used to assign faceted metadata records to documents. According to Pierre, the metadata records "serve as a bridge between a corpus of free text documents and a highly structured database with a rigid schema." They allow statistical techniques and traditional data mining algorithms to be applied to the set of structured metadata records to discover knowledge that otherwise is implicit in the underlying document collection.

One limitation of this technique is that it discovers associations from only the faceted metadata. The relationships among the actual concepts in the documents may not be captured all of the time.

2.3.2.3 Soft-Matching Association Rules Mining. Data mining algorithms can be easily influenced by variations and noises in text, such as morphological changes, typographical errors, misspellings, and abbreviations. Soft matching association rule mining takes into consideration of such variations, so that additional relationships can be discovered to more accurately reflect the regularities in the data (Nahm and Mooney, 2002).

The equality comparison method of the soft association rule mining makes it different from the normal association rule mining. In normal association rule mining, two items are equal only if there is an exact match. Such comparison cannot handle the heterogeneity of textual items. For example, "Windows 2000" and "Win2K" refer to the same object, but they are considered different if the exact match is used. Soft association rule mining introduces partial matching between textual items. Two items are considered equal if their similarity exceeds a predefined threshold. In such cases, the two items will

be treated as a single one to reflect the real granularity in the data. The similarity function thus is a key component in the soft association rule mining algorithm.

Similarity between two items can be calculated by two approaches: String-Edit Distance (Levenshtein, 1966) and Vector Space Model (Salton and McGill, 1983). String-edit distance is defined as the minimum number of insertions, deletions or substitutions necessary to transform one string into another. The less the number of operations is, the more similar the two items are. Vector space model treats the items as "bag-of-words," and a similarity measure, e.g. cosine measure, is calculated for a pair of vectors of items.

2.4 Interestingness Evaluation

Due to their unsupervised nature, association rules mining algorithms tend to produce a great many rules which can easily exceed the capability of a human user to comprehend the rules and identify interesting ones. There is a need for techniques that can identify interesting rules from the results according to the user's background and interests. Different approaches have been proposed to evaluate the interestingness of the discovered patterns, especially association rules, from different aspects, such as simplicity (size of rules), certainty (confidence), utility (support), and novelty (currently unknown). Evaluating the interestingness of a rule for a particular user requires a comparison of the rule to an existing body of knowledge the user is assumed to already possess. Overall, the evaluating techniques can be categorized into three categories: objective measures, subjective measures, and general or domain knowledge based measures.

2.4.1 Objective Measures

Objective measures depend only on the structure of the data and the characteristics of the extracted patterns. They can be handled with techniques requiring no user data and no application or domain knowledge.

2.4.1.1 Major Objective Measures. There have been many objective measures proposed to evaluate the interestingness of association rules. Thirteen measures for evaluating interestingness of association rules were theoretically and empirically compared in (Hilderman and Hamilton, 2001). Tan et al. (2002, 2004) conducted a similar comparison study with 21 objective measures, and described the key properties one should consider when selecting the right measure. The study by Tan et al. is by far the most comprehensive study on objective measure evaluation. Table 2.1 summarizes the 21 measures and their calculation formulas.

Table 2.1 Objective Interestingness Measures for Association Rules

#	Objective Measures	Calculation
1	Φ-coefficient	$\frac{P(A,B)-P(A)P(B)}{\sqrt{P(A)P(B)(1-P(A))(1-P(B))}}$
2	Goodman-Kruskal's (λ)	$\frac{\sum_j \max_k P(A_j,B_k)+\sum_k \max_j P(A_j,B_k)-\max_j P(A_j)-\max_k P(B_k)}{2-\max_j P(A_j)-\max_k P(B_k)}$
3	Odds ratio (α)	$\frac{P(A,B)P(\bar{A},\bar{B})}{P(A,\bar{B})P(\bar{A},B)}$
4	Yule's Q	$\frac{P(A,B)P(\bar{A}\bar{B})-P(A,\bar{B})P(\bar{A},B)}{P(A,B)P(\bar{A}\bar{B})+P(A,\bar{B})P(\bar{A},B)}=\frac{\alpha-1}{\alpha+1}$
5	Yule's Y	$\frac{\sqrt{P(A,B)P(\bar{A}\bar{B})}-\sqrt{P(A,\bar{B})P(\bar{A},B)}}{\sqrt{P(A,B)P(\bar{A}\bar{B})}+\sqrt{P(A,\bar{B})P(\bar{A},B)}}=\frac{\sqrt{\alpha}-1}{\sqrt{\alpha}+1}$
6	Kappa (κ)	$\frac{P(A,B)+P(\bar{A},\bar{B})-P(A)P(B)-P(\bar{A})P(\bar{B})}{1-P(A)P(B)-P(\bar{A})P(\bar{B})}$

Table 2.1 Objective Interestingness Measures for Association Rules (Continued)

#	Objective Measures	Calculation
7	Mutual Information (M)	$\frac{\sum_i \sum_j P(A_i,B_j)\log\frac{P(A_i,B_j)}{P(A_i)P(B_j)}}{\min\left(-\sum_i P(A_i)\log P(A_i), -\sum_j P(B_j)\log P(B_j)\right)}$
8	J-Measure (J)	$\max\left(P(A,B)\log\frac{P(B\mid A)}{P(B)} + P(A\bar{B})\log\frac{P(\bar{B}\mid A)}{P(\bar{B})},\right.$ $\left. P(A,B)\log\frac{P(B\mid A)}{P(B)} + P(A\bar{B})\log\frac{P(\bar{B}\mid A)}{P(\bar{B})}\right)$
9	Gini Index (G)	$\max\left(P(A)\left[P(B\mid A)^2 + P(\bar{B}\mid A)^2\right] + P(\bar{A})\left[P(B\mid \bar{A})^2 + P(\bar{B}\mid \bar{A})^2\right] - P(B)^2 - P(\bar{B})^2,\right.$ $\left. P(B)\left[P(A\mid B)^2 + P(\bar{A}\mid B)^2\right] + P(\bar{B})\left[P(A\mid \bar{B})^2 + P(\bar{A}\mid \bar{B})^2\right] - P(A)^2 - P(\bar{A})^2\right)$
10	Support (s)	$P(A,B)$
11	Confidence (c) *	$P(B\mid A)$
12	Laplace (L)	$\max\left(\frac{NP(A,B)+1}{NP(A)+2}, \frac{NP(A,B)+1}{NP(B)+2}\right)$
13	Conviction (V)	$\max\left(\frac{P(A)P(\bar{B})}{P(A\bar{B})}, \frac{P(B)P(\bar{A})}{P(B\bar{A})}\right)$
14	Interest (I)	$\frac{P(A,B)}{P(A)P(B)}$
15	cosine (IS)	$\frac{P(A,B)}{\sqrt{P(A)P(B)}}$
16	Piatetsky-Shapiro's (PS)	$P(A,B)-P(A)P(B)$
17	Certainty factor (F)	$\max\left(\frac{P(B\mid A)-P(B)}{1-P(B)}, \frac{P(A\mid B)-P(A)}{1-P(A)}\right)$
18	Added Value (AV)	$\max(P(B\mid A)-P(B), P(A\mid B)-P(A))$
19	Collective strength (S)	$\frac{P(A,B)+P(\bar{A}\bar{B})}{P(A)P(B)+P(\bar{A})P(\bar{B})} \times \frac{1-P(A)P(B)-P(\bar{A})P(\bar{B})}{1-P(A,B)-P(\bar{A}\bar{B})}$
20	Jaccard (ζ)	$\frac{P(A,B)}{P(A)+P(B)-P(A,B)}$
21	Klosgen (K)	$\sqrt{P(A,B)}\max(P(B\mid A)-P(B), P(A\mid B)-P(A))$

*: in the study by Tan et al. confidence is defined as max(P(A|B), P(B|A)). Here the standard definition in (Agrawal et al. 1993) is used. The details of each measure can be referred to in (Tan et al. 2004).

2.4.1.2 Classification of Objective Measures. According to the similarity in properties, Tan et al. (2004) grouped the major objective measures into seven groups, as shown in Table 2.2.

Table 2.2 Groups of Objective Measures with Similar Properties

Group	Objective Measures
1	Odds ratio (α), Yule's Q, Yule's Y
2	Cosine (IS), Jaccard (ζ)
3	Support (S), Laplace (L)
4	Φ-coefficient, Collective strength (CS), Piatetsky-Shapiro's (PS)
5	Gini Index (G), Goodman-Kruskal's (λ)
6	Interest (I), Added Value (AV), Klosgen (K)
7	Mutual Information (M), Certainty factor (F), Kappa (κ)

Table 2.3 Classification of Objective Measures

	Measures of deviation from equilibrium	Measures of deviation from independence
Descriptive measures	– confidence – Sebag et Schoenauer index, – example and counter-example ratio, – Ganascia index, – moindre-contradiction, – inclusion index...	– correlation coefficient, – lift, – Loevinger index, – conviction, – J-measure, – TIC, – odds ratio, – multiplicateur de cote...
Statistical measures		– implication intensity, – implication index, – likelihood linkage index, – oriented contribution to 2, – rule-interest...

Blanchard et al. (2005) identify two different but complementary aspects of the rule interestingness: the deviation from independence and the deviation from equilibrium (maximum uncertainty of the consequent given that the antecedent is true). An objective measure can also be either descriptive or statistical. Descriptive measures do not vary when all the data cardinalities are increased or decreased in equal proportion (cardinality

expansion); while statistical measures vary with the cardinality expansion. These two aspects classify the objective measures into a 2x2 table, as shown in Table 2.3.

Objective measures evaluate the interestingness of association rules from different perspectives. No single measure is better than others in all circumstances. The key properties of each measure need to be considered to select the right measure(s) for a given application (Tan et al. 2004).

2.4.2 Subjective Measures

While objective measures are important, subjective ones are ultimately needed to satisfy a particular user's needs (Liu and Hsu, 1996; Piatesky-Shapiro and Matheus, 1994; Silberschatz and Tuzhilin, 1995). Unlike objective measures, subjective measures depend on the specific needs and the prior knowledge of the user. Application/domain specific knowledge and/or the user's existing knowledge of the system are required to determine what is subjectively interesting to a user, but this is still considered a very difficult problem, especially for designing general and domain-independent subjective measures. Some of the reasons are: (1) in different domains users have different interests; (2) given the same application domain and data set, different users are interested in different subsets of the discovered rules; (3) for the same user, his/her interests may change over time. All these factors should be considered when subjective measures are used to identify interesting rules for particular users. Proposed subjective measures (or approaches to finding subjectively interesting rules) include interestingness constraints, unexpectedness and actionability.

2.4.2.1 Interestingness Constraints. The basic idea of using interestingness constraints is to limit the format and the content of the discovered rules to what is interesting defined by the user. Syntactic and semantic constraints are basic techniques to select interesting rules (Agrawal and Srikant, 1994). For example, the user can specify what types of terms should appear in a rule, or what particular values of a variable should be present. Using syntactic and semantic constraints to find interesting rules is straightforward and easy to implement, but it can only find what the user has expected, not the unexpected and unknown knowledge which sometimes is also useful.

Rule templates can be defined to limit the results to those interesting patterns that match the templates (Klemettinen et al. 1999). With templates, the user can specify both what is interesting and what is not. Interesting rules can be identified with inclusive template, while uninteresting ones are those that match exclusive template. Template is defined in the same format as that of association rules, except that the components in a template are class names, attribute names, instances of classes, and expressions of the three. Interesting and uninteresting rules are the instances of the templates. However, to be able to define such templates, users have to know what they do (or do not) want and the structure of the data set.

Metaquery (Shen et al. 1996) is a second-order predicates or template used to specify the forms of interesting rules. Metaquery presents a desired logic form of the rules to be discovered and serves as an important interface between human users and the discovery system. To extend the approach from single concept level to multiple concept level, Fu and Han (1995) proposed a meta-rule-guided data mining approach which applies meta-rules as guidance at finding multiple-level association rules. The predicates

in a meta-rule or template can be instantiated against the database schema, and some variables in the predicates can be bounded to multiple levels of concepts in the corresponding conceptual hierarchies.

2.4.2.2 Unexpectedness. Unexpectedness measures are developed upon the assumption that patterns are interesting if they are unexpected or previously unknown to the user (Frawely et al. 1991). Most of the existing approaches measure unexpectedness by comparing the discovered rules to the users' expectation and unexpectation. The user is required to explicitly express what he or she thinks is interesting and not interesting using a defined language and grammar.

In (Liu et al. 1999), unexpectedness of the rules to a user is determined by asking the user to specify a set of patterns according to his/her previous knowledge or intuitive feelings. The specified set of patterns is then used in a fuzzy matching algorithm to match and rank the discovered patterns. The user-expected patterns are described with the fuzzy linguistic variables. A fuzzy pattern matching method is implemented to compare each pattern in the discovered set to the user's specifications. The matching technique should be tailored for different types of patterns. For example, matching classification rules should be different from matching association rules. The degrees of match between the discovered patterns and the user specifications are used to rank the discovered patterns.

This approach is further extended in the IAS system (Liu et al. 2000), in which the user's expectation is divided into three types: general impressions, reasonably precise concepts and precise knowledge, and defined with a given specification language. The assumption behind these techniques is that the domain knowledge or users' interests are

buried in the user specified patterns. However, in most situations, users do not know what to expect from a text mining system. Even if they do, it is difficult to explicitly express the expectations with a given language in a given format.

A belief-driven method is also proposed for discovering unexpected patterns (Padmanabhan and Tuzhilin, 1998; Padmanabhan and Tuzhilin, 1999). Prior background knowledge of the user constitutes a set of expectations or beliefs that user has about the problem domain, and is used to seed the search for patterns that contradict the beliefs.

2.4.2.3 Actionability. Actionability measures are developed upon the assumption that "rules are interesting if the user can do something with them to his/her advantage" (Liu et al. 2000; Piatesky-Shapiro and Matheus, 1994). Actionable rules are expected to be useful because the user can benefit from them by taking appropriate actions with the discovered knowledge. However, actionability is an elusive concept, because there are no predefined actions attached to a fixed space of rules. Actionability is not mutually exclusive with unexpectedness, as they complementarily measure the interestingness of discovered patterns. It is possible that (1) rules that are both unexpected and actionable; (2) rules that are unexpected but not actionable, and (3) rules that are actionable but expected (Silberschatz and Tuzhilin, 1996).

Similar approaches to unexpectedness identification can be applied to find actionable rules. The user first needs to specify all possible actions he/she can take, and then for each action gives the situations under which this action will be taken. The situations are represented by a set of user-specified action patterns, against which the discovered patterns will be matched to find the actionable ones.

An opposite way is to identify non-actionable association rules, so that the number of rules the user has to examine for useful rules can be reduced (Liu et al. 2001). Unlike other pruning techniques that use general rules (with fewer conditions) to prune insignificant specialized rules (with more conditions) and remove insignificant and redundant rules, identification of non-actionable rules analyzes rules backwards, i.e. using higher quality specialized rules to determine whether a more general rule is potentially actionable. It partly solves the problem that significant and/or non-redundant rules may not be useful for action.

2.4.3 General or Domain Knowledge

This approach is in between the usage of objective and subjective measures. It measures the interestingness of a rule by comparing it to some well-known general or domain knowledge.

WordNet, which is a general lexical database (Fellbaum, 1998), is used to evaluate the novelty of extracted association rules (Basu et al. 2001). Novelty is measured as the semantic distance between two words based on the length of the shortest path connecting them in the WordNet hierarchy and the number of direction changes of relations along the shortest path. The more direction changes in the path from one word to another, the greater the semantic distance between the words, since changes of direction along the path reflect large changes in semantic context. The novelty of a rule is then defined as the average value of the distances across all pairs of words with one in the antecedent and one in the consequent of the rule.

The evaluation shows that the WordNet novelty measure has a high correlation with the novelty ratings by human users. However, because WordNet is a general lexical

database, it does not differentiate users with different backgrounds. A rule that is novel to one user may not be novel to others.

2.5 Summary

The literature review on text mining suggests that association rule mining is an important technique to find associations between items, and it can be applied to textual data as well. Some of the methods use the existing document features, such as author assigned keywords and document category labels, to discover associations, but they are limited to documents that already have predefined features only. Other methods apply document classification techniques to assign documents with predefined category labels, and discover associations among these labels. These methods can be applied to documents without predefined features, but, similar to the first type of methods, they cannot discover associations between concepts that appear in documents but are not selected as document features. On the other hand, information extraction techniques have been developed to extract salient information from documents. For example, a noun phrase extractor can identify noun phrases that usually carry the primary information of a document. Combining these extraction techniques with data mining algorithms could be a solution for knowledge discovery from text.

When the number of discovered patterns become too large for users to find interesting ones easily and quickly, interestingness measures are needed to evaluate the interestingness of the patterns. Interestingness measures can be classified from two dimensions: user-effort and user-orientedness. User-effort is the amount of work a user has to complete in order to generate the measure; user-orientedness is the degree to which

a measure is customized for a particular user by considering his/her background knowledge or interests. Figure 2.4 shows different measures along the two dimensions.

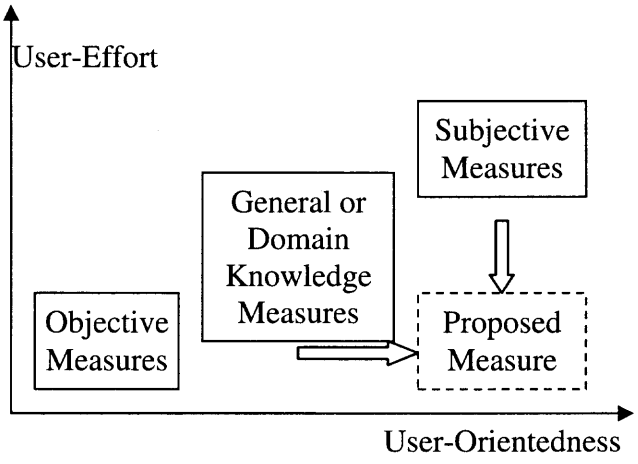


Figure 2.4 Interestingness measures.

Objective measures involve low user-effort because they can be completely generated from the rules and the underlying data, but they are not user-oriented. General or domain knowledge measures require middle-level user-effort because the general or domain knowledge is usually developed by human experts. Though they can customize the results for all users, they do not differentiate users with different backgrounds and interests. Subjective measures are able to evaluate the rules for a particular user, but they usually require high user-effort, e.g., asking the user to explicitly express his/her expectations about the results. This study proposes an interestingness measure that requires low user-effort but is highly user-oriented. The objective is achieved by learning the implicit user's background knowledge and interests from documents and applying them in the knowledge discovery process.

CHAPTER 3

DISCOVERING NOVEL ASSOCIATION RULES FROM TEXT

The literature review demonstrates that users' background knowledge and interests are important in text mining process for discovering interesting patterns, but such knowledge cannot be easily obtained if the user has to express it explicitly. The interestingness measures that take account of users' background knowledge without too much user-effort are needed.

This chapter presents the methodology for discovering association rules from text and the user-oriented novelty that measures the interestingness of discovered rules. It starts with an overview of the proposed methodology, including the basic notions used throughout the dissertation, the user-oriented text mining framework, the assumptions and the scope of this study. The algorithms for discovering novel association rules are then described in detail, including the background knowledge learning algorithm, document feature extracting algorithm, association rule mining algorithm and user-oriented novelty calculation algorithm.

3.1 Overview

The overview presents a brief introduction to the research, including the basic notions used throughout the dissertation, the user-oriented text mining framework, the assumptions and the scope of the study.

3.1.1 Definitions

Some key terms used throughout the dissertation are defined as follows.

Stop words. Stop words are commonly used words, such as *the*, *is*, *about* and *than*. They are functional and not content bearing. A complete stop word list used in this study can be found in Appendix A.

Keywords. Keywords are content bearing and non-functional. They are the words that are not on the stop word list, and they are single words as compared to phrases. It is necessary to address the distinction between *keywords* used in this study and those in most academic papers. In this paper, a keyword refers to a single word that appears in a document but not on the stop word list. In academic papers as well as many other articles, keywords are a few phrases assigned by the author(s) to identify the main topics of an article or the major categories to which an article belongs. They often appear at the end of the abstract of the article.

Noun phrase. A noun phrase is a group of words that has the same syntactic role as a noun in a sentence. There are two types of noun phrases: base noun phrases and complex noun phrases. The former refers to simple and non-recursive noun phrases that do not contain other noun phrase descendants. They are non-overlapping segments of a sentence. The latter is recursive and overlapping, and can be further segmented into more base and/or complex noun phrases. In this study, only base noun phrases are extracted.

A noun phrase is formally defined as $A^* N+$, where A refers to an adjective, N refers to a noun, * means none or more instances, and + means one or more instances. The pattern defines a base noun phrase that consists of at least one noun proceeded by none or more adjectives. Examples are *rule*, *association rule*, and *novel association rule*.

Concept hierarchy. Concept hierarchy is defined as a directed acyclic graph consisting of concepts extracted from documents. This hierarchical structure has the following characteristics:

- A parent node would refer to a more general concept than its children, in other words, the parent's concept subsumes the child's (Sanderson, 1999).
- There is one and only one root item.
- Each node can have none or more children.
- Except the root that does not have any parent, each node can have one or more parents. This is different from a standard tree in which every node except the root has one and only one parent. A node has more than one parent when the corresponding term has multiple senses in all documents, meaning that it can appear in multiple positions in the concept hierarchy.

Figure 3.1 shows an example of a concept hierarchy taken from (Sanderson, 1999), with additional elements (the root *Economy*) and relationships (*Automobile* → *Passenger automobile; Passenger automobile* → *Car*) added to illustrate the hierarchy characteristics discussed above.

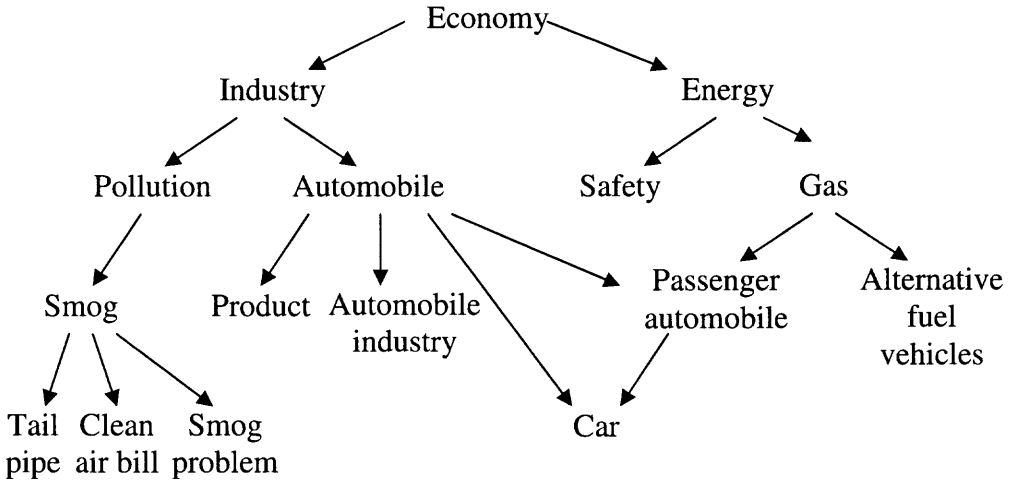


Figure 3.1 An example of concept hierarchy.

Background documents. Background documents are documents that are already known to a user. They are related to a certain topic the user is interested in or a certain domain the user is working in. In this study, users will provide their background documents they have read for a certain type of tasks they are performing, e.g. academic research. The relevancy judgment is made by the user; in other words, when the user submits a document to the system, it is assumed relevant to the user's background.

Target documents. Target documents are selected from a large corpus. The text mining algorithms will be performed on the target documents rather than the original corpus, because the corpus may contain documents with heterogeneous topics, and many of them may not be relevant to the user's background.

Target documents can be automatically generated with information retrieval techniques. A query can be formulated from the user's background knowledge or interests, and executed to retrieve relevant documents from the corpus. Boolean keyword matching or different similarity measures can be used to judge the relevance of a document in the corpus to the user's interests or background.

Interestingness measures. Interestingness measures are developed with the goal of discovering only the patterns that are interesting to a particular user. Such measures evaluate the discovered patterns from a certain perspective and try to predict the interestingness of a rule with a score. Section 2.2 gives details of interestingness measures and their classifications.

Novelty. Novelty is one of the perspectives from which the interestingness of a rule can be evaluated. It identifies potentially interesting rules by predicting to what degree a rule is currently unknown to the user. Novelty is similar to the unexpectedness

measure proposed in other studies, but is measured differently. Evaluating the novelty of a rule requires the comparison of it to an existing body of knowledge the user is assumed already to possess (Basu, 2001). In this study, the distance between the antecedent and the consequent of a rule in the background knowledge is defined as its novelty. Because the proposed novelty measure is directly associated with the user's background knowledge, it is called user-oriented novelty measure.

Usefulness. Though novel (unknown) rules seem interesting, they are not guaranteed to be useful. Interestingness measures may be able to identify interesting rules, but whether or not the identified rules are useful is still undetermined. In this study, rule usefulness is judged by the users. Different interestingness measures, including the proposed user-oriented novelty measure, are compared with the subjective usefulness ratings to evaluate the performance of the interestingness measures in terms of identifying useful rules.

3.1.2 The User-oriented Text Mining Framework

Most of the existing text mining approaches take a data-centered view, in which the result is solely determined by the data set. It has been realized that users' background knowledge, interests and information needs are also very important in text mining tasks. Such user-related variables should be deployed in the text mining process to further refine the results, and they could also be used to develop certain measures to evaluate the interestingness of the results.

Taking into consideration of users' background information requires a user-centered view of text mining. In this study, a user-oriented text mining framework is proposed, as shown in Figure 3.2.

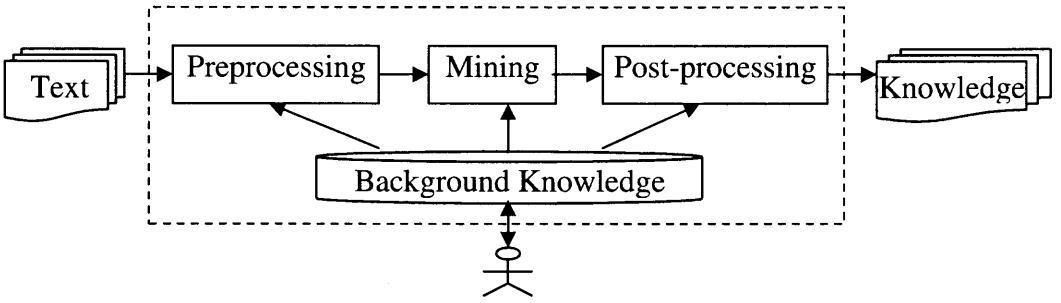


Figure 3.2 The user-oriented text mining framework.

The mining process follows the standard steps: preprocessing, mining, and post-processing. An additional component, background knowledge developer, is added. The background knowledge developer considers the user's current knowledge or interests, collects materials that can best represent such kind of knowledge or interests, and develops a certain data structure from the features extracted from the materials. The data structure, in turn, interferes with each step of the mining process.

During the preprocessing stage, the background knowledge could be used to select relevant documents from the original document set and to clean the extracted features. Background knowledge is essential in the mining stage. It enables the discovery of patterns that are potentially interesting to a user by comparing a candidate pattern with the user's background knowledge. The comparison might be delayed to the post-processing stage, since the delay enables the user to examine the rules manually and interpret the results from different angles.

3.1.3 Deployment of the Framework

The proposed framework is applied to the problem of discovering novel association rules from text, with the presence of user's background knowledge derived from a set of background documents. The system models the user's background knowledge as a

concept hierarchy consisting of keywords extracted from the user's background documents, and measures the novelty of an association rule as the semantic distance between the antecedent and the consequent of a rule in the background knowledge.

The discovery process is divided into two major stages: background knowledge development and novel association rules discovery. Background knowledge is developed from a set of documents, namely background documents, which are already known to the user. Keywords are extracted from background documents to form a keyword space. Popular keywords are organized into a hierarchical structure by using the POCA (Probability of Co-occurrence Analysis) technique. The keyword space containing a concept hierarchy is called the background knowledge, which captures the semantic usage of keywords in the background documents. Target documents are collected from a large corpus by selecting documents relevant to the user's background. Association rules are mined among noun phrases extracted from target documents. User-oriented novelty of a rule is defined as the semantic distance between the antecedent and the consequent of a rule in the background keyword space.

Different units (keyword in background knowledge development vs. noun phrase in association rules mining) are used in the two stages because of the following reasons:

- Indexing documents with words is a proven technique in information storage and retrieval (Salton and McGill, 1983). It is efficient and computationally inexpensive. The purpose of developing the background knowledge is to model the background documents in a way such that later comparison with other objects (e.g., target documents, association rules, etc.) is possible. Though words are less descriptive than phrases, the bag-of-words approach has successfully modeled the documents in information retrieval applications. Moreover, the background knowledge is transparent to the user and there is no need for the user to interpret it.

- The association rules are presented to the user, so they must be descriptive and interpretable in order to be useful. Phrases are more descriptive than single

words, and noun phrases carry the primary information of a document. Therefore, it is a better choice to use noun phrases for association rule mining. In addition, using phrases as features can avoid mining such rules that are resulted from phrase usage (e.g. *Wall -> Street*).

- Though using phrases to develop the background knowledge may result in a more interpretable concept hierarchy, it becomes difficult to compare the discovered knowledge to the background knowledge in order to calculate a distance measure. Phrase matching is more difficult than word matching, because there could be more variations in phrase expressions. Different phrases may refer to the same concept, and phrases with the same noun head may refer to different concepts. A special circumstance in which a phrase needs to be compared with its sub-phrases also needs to be considered. The usage of words in background knowledge makes it easier to calculate the distance between other objects that are composed of words.

3.1.4 Differences from Existing Approaches

Though this study borrows some ideas from existing approaches, there exist major differences. The major characteristics of the proposed method are that (1) the background knowledge derivation is automatic, and (2) a user-oriented novelty measure is developed to predict the interestingness of the discovered rules by measuring the semantic distance between the left and the right side of a discovered rule in the background knowledge. The approach differs from existing approaches in that:

- The user-oriented novelty measure is particular for a user or a group of users. Using a general purpose knowledge database, such as WordNet, to evaluate the novelty of the rules (Basu et al. 2001), cannot differentiate users with different backgrounds.

- The background knowledge is derived from a set of background documents. Users are not asked to explicitly express their existing knowledge or expectations. It is difficult for users to do that, and the expression may be limited to what the user can think of at that particular time. The proposed approach only asks users to provide a set of documents that they have read and identified as relevant to their interests. Deriving the background knowledge from a set of documents not only reduces the user's workload but also captures the background knowledge as much as possible.

- Unlike other systems, such as IAS (Liu et al. 2000), the proposed system does not require the user to learn a specification language as well as the grammar. The background knowledge derivation is transparent to users. Users only need to provide documents that they think are relevant to their interests.

Even if the user is able to specify expected patterns as described in (Liu et al. 1999), the number of expectations is usually too small when compared with the huge number of rules produced by a text mining system. Therefore, many "interesting" patterns that are not really interesting could be discovered just because they are not covered in the specified expectations. The proposed approach overcomes this problem by developing a concept hierarchy that captures as many relationships as possible between keywords in the background documents.

3.2 Discovering Novel Association Rules

In the overview, the user-oriented text mining framework has been discussed. This section presents the application of this framework to a specific text mining problem, namely discovering novel association rules from text. The discovery process includes three steps: background knowledge development, association rules mining, and novelty evaluation.

3.2.1 Background Knowledge Development

Evaluation of the novelty of discovered association rules requires comparing the rules to the knowledge body the user possesses. There are various ways to obtain the knowledge the user is assumed to possess. The most straightforward way is to ask the user to express what he/she knows or wants, so called expectations (Liu, 1999). Another way is to use general (e.g., lexical database) or certain domain knowledge (e.g., domain-specific

ontology). These approaches are either difficult for users to use, or insufficient to represent users' real knowledge.

In this study, an algorithm for implicitly deriving the user's background knowledge from a set of background documents is proposed. Keywords are extracted from the background documents and a keyword space is constructed. The POCA technique is then used to develop a concept hierarchy which organizes the popular keywords in a hierarchical structure. The selection of popular keywords is determined by a *document frequency* threshold. Only the keywords whose *document frequency* is larger than a given threshold are chosen to appear in the concept hierarchy. The keyword space containing the concept hierarchy models the user's background knowledge.

3.2.1.1 Keyword Extraction. A stop word list is used to remove all functional and non-content-bearing words (stop words, see Appendix A for details of the stop word list), and the remaining words are considered keywords. Keywords are then converted to their root forms by eliminating the inflectional morphological variants (e.g. plurals of noun) (Kantrowitz et al. 2000). The conversion enables more accurate frequency count of keywords.

3.2.1.2 Keyword Indexing. The unique keywords from all background documents are sorted into a unified list, and each node in the list corresponds to a unique keyword in its root form and its frequency, as well as a pointer to a list of documents where the keyword occurs. The frequency of the keyword in each document is also saved. The index structure is denoted in Figure 3.3.

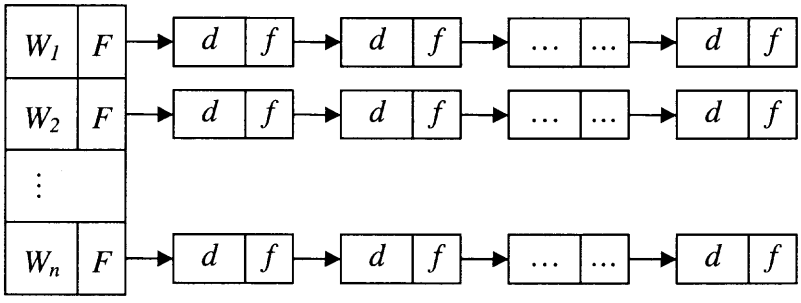


Figure 3.3 Keyword index.

W represents a keyword, and *F* is its frequency in the background document set. *F* counts all the occurrences of *W* in all documents, while *f* is the frequency of the keyword in a particular document, whose identification number is *d*. This index structure enables a quick lookup of a keyword, its total frequency and its frequency in a particular document. It also allows a convenient calculation of a keyword's document frequency (the number of documents it occurs in). The index structure is also called an inverted file, and can be saved into a file on the hard disk.

3.2.1.3 Concept Hierarchy Development. The concept hierarchy captures the semantic usage of keywords and their relationships in the background documents. Different approaches have been proposed to derive a hierarchical structure to organize the features extracted from documents.

In information retrieval, the generality and specificity of terms are measured by their document frequency (DF). The more documents a term occurs in, the more general it is. Forsyth and Rada (1986) introduced the use of DF to derive a multi-level structure that has general terms on top of specific terms. Sanderson and Croft (1999) applied this idea to build and present concept hierarchies derived from text by using subsumption to create a topic hierarchy. For two terms, *X* and *Y*, *X* is said to subsume *Y* if $P(X|Y) >= N$,

$P(Y|X) < 1$, where $P(X|Y)$ is the conditional probability that X appears in a document, given Y also appears in that document.

However, in some cases, subsumption might yield term pairs where X does not subsume Y. For example, $P(X|Y)=0.8$ and $P(Y|X)=0.9$. To overcome this problem, Wu (2001) developed a revised subsumption called Probability Of Co-occurrence Analysis (POCA). It is re-defined as $P(X|Y) > P(Y|X)$, $P(X|Y) >= N$, where $0 < N <= 1$. If a term pair (X, Y) fulfills the above set of inequalities, X is the parent of Y. Note, the threshold, N, affects the number of term pairs derived; namely, a larger N results in a smaller number of term pairs. In both Sanderson's and Wu's studies, a threshold $N=0.8$ was used, because it yielded better results. Another threshold, document frequency, is also useful to select terms that are significant (whose document frequency is equal to or greater than the given threshold). The POCA technique is used to develop a concept hierarchy to organize the keywords extracted from background documents. After the keywords are extracted and the concept hierarchy is developed, the background knowledge is constructed. The background knowledge, represented by a keyword space with a concept hierarchy inside, is shown in Figure 3.4.

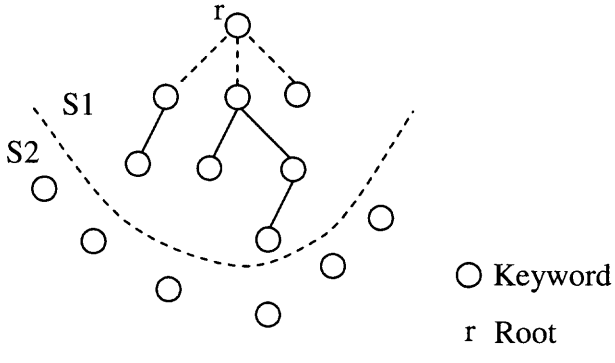


Figure 3.4 Background knowledge keyword space.

Keywords are shown as circles. The keyword space is divided into two areas: S1 and S2. S1 contains keywords that are included in the hierarchy, and S2 contains keywords that are not because they do not satisfy the *DF* constraint. A virtual keyword, *r*, is introduced as the root to connect all first-level keywords in the hierarchy. Connections between the root and the first-level keywords are represented as dotted lines.

3.2.2 Target Documents Selection

Target document selection can be viewed as a document retrieval process, in which a query can be formulated from the user's interests or derived from the background documents by selecting the most representative keywords from the keyword space. The query can be issued to the search system of a large document collection to retrieve relevant documents. Whether a document is relevant or not is determined by the similarity between the query and the document. There are various approaches to calculate the similarity between a query and a document. Below is the description of the approach that uses the Vector Space Model (VSM) (Salton and McGill, 1983) to model both the query and the documents and the cosine similarity to measure the relevancy of the document to the query.

3.2.2.1 Vector Space Model. The Vector Space Model is one of the most well known best-match models in information retrieval. In the vector space model, a vector is used to represent each document in a collection. Each component of the vector corresponds to the index unit (word, phrase, or term) associated with a given document. The value assigned to that component reflects the importance of the unit in representing the semantics of the document. Typically, the value is a function of the frequency with which the term occurs in the document or in the document collection as a whole (Dumais,

1991; Jones, 1972). For example, if a document is described for indexing purposes by the three terms *text*, *mining* and *algorithms*, it can then be represented by a vector in the three corresponding dimensions. If a weighting function assigns weights 0.5, 2.5, and 5.0 to the three terms respectively, the word *algorithms* is considered the most significant term in the document, with *mining* of the secondary importance and *text* of the least importance.

The TF.IDF (Term Frequency, Inverse Document Frequency) weighting scheme (Salton and Buckley, 1988) is often used to assign larger weights to terms with higher discrimination power in a document. It makes two assumptions about the importance of a term in a particular document. First, the more often it appears in the document (term frequency), the more important it is for the document. Second, the more often it appears across the entire collection of documents (inverse document frequency), the less important it is for the document since it does not distinguish the document well from others. In the TF.IDF framework, the weight of term t_j in a document d_i, $w_{ij,}$ is defined as

$$w_{ij} = tf_{ij} \times \log_2 \frac{N}{n} \tag{3.1}$$

where tf_{ij} is the frequency of term t_j in document d_i, N is the total number of documents in the collection, and n is the number of documents where term t_j occurs at least once. With the vector space model, each document is modeled as a vector of weights. It is denoted as

$$d_i = \left(w_{i1}, w_{i2}, \ldots w_{ij}, \ldots w_{in}\right) \tag{3.2}$$

where d_i is the i_{th} document, n is the total number of unique keywords, and w_{ij} is the weight of the j_{th} keyword in document i.

3.2.2.2 The Matching Process. Matching a query to documents is the fundamental process in information retrieval. When the query and the document representations are similar, the query can be considered as a point in the document space. In such case, relevant documents are those near the query point in the document space. When a vector space model is used, distance between a query and a document is calculated by comparing their vectors using similarity measures, such as distance or cosine correlation. The assumption is that the more similar a document vector is to a query vector, the more likely that the document is relevant to that query.

Calculation of the distance or cosine correlation is straightforward when the vector space model is used to determine the similarity between two documents. For the cosine correlation measure, the similarity between two documents x and y is the cosine of the angle between two vectors $\vec{x}$ and $\vec{y}$ representing x and y respectively, and calculated by the formula

$$Similarity(x, y) = \frac{\vec{x} \cdot \vec{y}}{|\vec{x}| \times |\vec{y}|} \tag{3.3}$$

where $|\vec{x}|$ and $|\vec{y}|$ are the norms of the two document vectors. The similarity between two documents can also be measured by their distance in the document space. The most commonly used distance measure is Euclidean distance, which is calculated as

$$Euclidean(x, y) = \sqrt{\sum (\vec{x}_i - \vec{y}_i)^2} \tag{3.4}$$

Given two documents d_i and d_j and their vector representations in formula (3.2), the cosine similarity between them is denoted as

$$Similarity(d_i, d_j) = \frac{d_i \cdot d_j}{|d_i| \times |d_j|} = \frac{\sum_k w_{ik} w_{jk}}{\sqrt{\sum_p w_{ip}^2 \sum_q w_{jq}^2}} \tag{3.5}$$

Once the similarity between the query and each document is calculated, the document list can be sorted by the similarity score. Target document set can be formed by selecting the documents whose similarity scores are above a defined threshold, or by selecting a certain percentage or number of documents from the top of the list.

3.2.3 Feature Extraction from Target Documents

Feature extraction extracts elements of interest from documents, so that the unstructured documents can be converted to structured feature sets, upon which the mining algorithms can be performed. Meaningful and descriptive document features can lead to easy-understanding and more potentially useful rules. The evidences from language learning of children (Snow and Ferguson, 1997) and discourse analysis theories, e.g. Discourse Representation Theory (Kamp, 1981), show that the primary concepts in text are carried by noun phrases. Also, noun phrases in documents are more descriptive than single words. Therefore, noun phrases are extracted as document features from target documents.

Noun phrase extraction includes two steps: part-of-speech (POS) tagging and noun phrase identification. POS tagging is to assign the right POS tag to each token in the text. Once each token in the free text is tagged, noun phrases can be identified by selecting the sequence of words whose POS tags are of interests. The part-of-speech tagger used in this study is a revised version of the widely used Brill tagger (Brill, 1992; Brill, 1995). The Brill tagger is based on transformation-based error-driven learning. When it was trained on 600,000 words from Wall Street Journal Corpus, an accuracy of 97.2% was achieved on a separate 150,000 words test set from Wall Street Journal Corpus. In this study the tagger was trained on two corpora, the Penn Treebank Tagged

Wall Street Journal Corpus and the Brown Corpus. Tagging is done in two stages. First, every word is assigned with a most likely tag. Next, contextual transformations are used to improve accuracy.

After all the words in the document are tagged, the noun phrase extractor extracts noun phrases by selecting the tokens whose POS sequence matches the predefined patterns. Only base noun phrases are used in this study, and their sequence pattern is defined as A* N+, where A refers to adjective, N refers to noun, * means none or more instances, and + means one or more instances. A set of optional rules is also used to filter out unwanted noun phrases. For example, users may choose to extract noun phrases with a certain minimum length (number of words).

The TF.IDF term weighting scheme is applied to reduce the number of features in each target document, and, more importantly, to select the significant features from each target document. Only those noun phrases whose weights are greater than a given threshold are selected as document features. After the extraction of noun phrases, each target document is converted into a vector of noun phrases, and the target document set becomes a structured record set of vectors of noun phrases.

3.2.4 Association Rules Mining

After feature extraction, target documents are converted into structured vectors of noun phrases. The standard APRIORI algorithm (Agrawal, 1993; Agrawal, 1994) is applied to identify the frequent noun phrase sets and the association rules among noun phrases.

A formal model of association rules mining between noun phrases is denoted as: Let $K = \{k_1, k_2, \ldots, k_m\}$ be the noun phrase set. Let D be the document set. Each record in D is a document d, which is represented as a binary vector, with $d[k]=1$, if d contains

noun phrase k, and $d[k]=0$ otherwise. Let X be a set of noun phrases. Document d satisfies X only if for all noun phrases k_i in X, $d[k]=1$. The problem of association rule mining is to generate all rules that are in the expression of $X \rightarrow Y$ and satisfy two forms of constraints - *support* and *confidence*. *Support* constraints specify the minimum number of documents in D that should satisfy the set of noun phrases in the rule. It is defined to be the fraction of documents in D that satisfy the union of noun phrases in X and Y. *Confidence* is the fraction of documents satisfying X that also satisfy Y, i.e. the conditional probability of Y given X in the database.

3.2.5 Interestingness Evaluation

The number of discovered association rules is usually too large for a user to look for interesting rules quickly and easily. An interestingness measure is needed to rank the rules so that it is easier for the user to find interesting results. Basu et al. (2001) use the WordNet lexical database to evaluate the novelty of association rules by calculating the semantic distance between two words in the WordNet hierarchy. WordNet, however, is a general lexical database and does not differentiate users with different backgrounds. In this study, the keyword space containing a concept hierarchy developed from the background documents at the early stage is used to measure the novelty of extracted association rules.

3.2.5.1 Definition of the User-oriented Novelty. The user-oriented novelty of an association rule is defined as the semantic distance between the antecedent and the consequent of the rule in the background knowledge. The distance between two itemsets is defined as the average of the distances between all term pairs, each of which consists of one term from the antecedent and one from the consequent of the rule. For example,

given a rule *[A, B] -> [C, D]*, its novelty is calculated as *average(D(A,C), D(B,C), D(A,D), D(B,D))*, where *D(X,Y)* is the semantic distance between items *X* and *Y*. Therefore, the problem of calculating the novelty measure can be transformed to the calculation of the semantic distance between two keywords in the background knowledge keyword space. The semantic distance between two keywords in the background knowledge is measured from two perspectives – occurrence distance and connection distance.

3.2.5.2 Occurrence Distance. Occurrence distance measures how distinct the occurrences of two keywords are in the background documents. Given two keywords *X* and *Y*, the more often they co-occur, the less the occurrence distance is. Keywords with less occurrence distance could be synonyms or highly interdependent terms. For instance, in data mining research papers about association rule mining, *support* and *confidence* tend to appear together very frequently. Synonyms are also possible when an author uses different words but refers to the same concept to avoid repetition. A large distinction in the occurrences of *X* and *Y* indicates a less strength of association between *X* and *Y*.

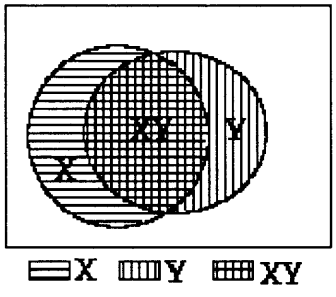


Figure 3.5 Occurrence distance.

Figure 3.5 shows the occurrence distance between *X* and *Y*. Given the probability that *X* and *Y* co-occur *P(XY)*, the distinction between the occurrences of *X* and *Y* is

P(XUY)-P(XY), where *P(XUY)* is the probability of the joint occurrence of *X* and *Y*. If the occurrence distance is normalized by the joint occurrence, the occurrence distance can be denoted as

$$D_o(X, Y) = (P(XUY)-P(XY))/P(XUY) = 1-P(XY)/P(XUY), \quad (3.6)$$

where $D_o(X, Y)$ is the occurrence distance between *X* and *Y*, *P(XY)* is the probability that *X* and *Y* co-occur, and *P(XUY)* is the probability that *X* or *Y* occurs. Occurrence distance ranges from 0 to 1. When two keywords do not co-occur, their occurrence distance is 1; when they have the exact same occurrence, the occurrence distance is 0.

3.2.5.3 Connection Distance. Connection distance measures the strength of the connection between two keywords in the concept hierarchy. If the occurrence distance measures the direct relationship between two keywords, the connection distance measures their relationships by considering their connections to other common keywords in the concept hierarchy. It is possible that two keywords may still have a certain relationship even if they do not co-occur in the background documents. Such a relationship can be captured by the connection distance measure.

For instance, *bird* and *fish* may not appear in the same documents often, but if they both co-occur with *animal,* in the concept hierarchy there will be two paths: *animal -> bird* and *animal -> fish.* The common keyword *animal* in the two paths suggests that there is a relationship between *bird* and *fish*, and this relationship is captured by the connection between *bird* and *fish* in the concept hierarchy: *bird – animal – fish.* The strength of the relationship could be reflected by the length of the connection path. The

longer the path is, the weaker the connection is. Figure 3.6 shows the connection distance of *X* and *Y* through their connections with *Z*.

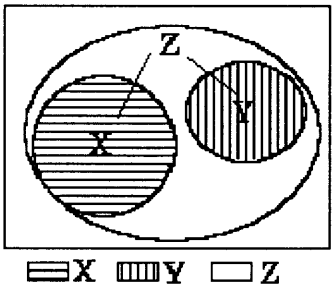


Figure 3.6 Connection distance.

In other words, connection distance measures how far apart two keywords are in the concept hierarchy, so it is also called hierarchy distance, which is defined as the length (number of keywords) of the shortest path connecting *X* and *Y* in the concept hierarchy and denoted as $d_h(X, Y)$. It is normalized by the maximum hierarchy distance between any two keywords. The normalized hierarchy distance is denoted as $D_h(X, Y)$. Calculation of the hierarchy distance $d_h(X, Y)$ and the maximum hierarchy distance is described in details as follows.

For easy explanation, keywords are classified into two categories: background keywords and target keywords. The former refers to those keywords that appear in background documents, and the latter refers to the keywords that occur in target documents. The two types of keywords are overlapped, since a number of keywords can appear in both background and target documents. The keywords and their possible locations in the keyword space are shown in Figure 3.7.

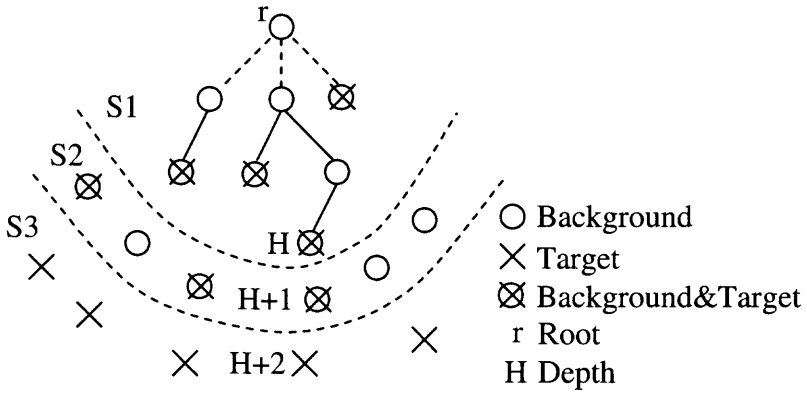


Figure 3.7 Hierarchy distance calculation.

In Figure 3.7, background keywords are shown as circles, and target keywords are displayed as crosses. Three areas, S1, S2, and S3, contain different types of keywords. S1 and S2 contain all background keywords. Because a document frequency threshold is applied when the concept hierarchy is developed, background keywords whose document frequency is less than the threshold are not present in the concept hierarchy. These keywords fall into area S2, and other keywords are placed in the hierarchy in area S1. Target keywords could fall into any one of the three areas. Those that are not found in background keyword space (areas S1 and S2) form area S3, while others are in either S1 or S2. In Figure 3.7 the circle-cross symbol represents keywords that are in both background and target documents.

r, the root of the hierarchy, is a virtual keyword introduced to connect all first level keywords in the concept hierarchy. Such connections are represented as dotted lines in Figure 3.7. H is the depth of the hierarchy, which is the number of words in the longest path from the root r to any leaf in the concept hierarchy. Calculation of the hierarchy distance between two keywords X and Y in different areas is discussed as follows.

There are two basic conditions: X and Y are both found in the concept hierarchy, and otherwise. In the first condition, the paths between X and Y in the hierarchy are identified, and the shortest one is selected and its length (the number of words in the path including X and Y) is assigned as the hierarchy distance between X and Y, $d_h(X, Y)$. The reason to select the shortest path is that the concept hierarchy development algorithm allows a keyword to be placed in multiple positions, and there could be multiple paths between two keywords. The selected path may include the root. In such case a penalty of 1 is added to the length of the path because the root is not an actual keyword. In the second condition, X and Y are not both present in the concept hierarchy, so there is no real connection between them. In such cases, $d_h(X, Y)$ is defined as the sum of $d_h(r, X)$ and $d_h(r, Y)$. The hierarchy distance between a keyword W and the root r is calculated as (1) the length of the path between W and r if W is in S1, or (2) H+1 if W is in S2 (because the distance should be larger than the largest distance between any keyword in S1 and r, which is the hierarchy depth H), or (3) H+2 if W is in S3. The hierarchy distance is maximized when two keywords are both in S3. According to the definitions given above, the maximum hierarchy distance is 2(H+2). Calculation of the hierarchy distance $d_h(X, Y)$ in different conditions is summarized in Table 3.1 (Len(X-Y) is the length of the shortest path between X and Y in the hierarchy).

Table 3.1 Hierarchy Distance Calculation

Y \ X	S1	S2	S3
S1	Len(X-Y) or Len(X-Y)+1	Len(X-r)+(H+1)	Len(X-r)+(H+2)
S2	Len(Y-r)+(H+1)	2(H+1)	(H+2)+(H+1)
S3	Len(Y-r)+(H+2)	(H+2)+(H+1)	2(H+2)
Len(X-Y) is the length of the shortest path between X and Y in the hierarchy			

After the hierarchy distance $d_h(X, Y)$ and the maximum hierarchy distance are calculated, they can be used to calculate the normalized hierarchy distance $D_h(X, Y)$.

3.2.5.4 Novelty Calculation. The semantic distance between two keywords X and Y is defined as the square root of the product of their occurrence distance and hierarchy distance, which is denoted as follows

$$\begin{aligned} D(X,Y) &= \sqrt{D_o(X,Y)\cdot D_h(X,Y)}, \\ D_o(X,Y) &= 1 - P(XY)/P(X \cup Y), \\ D_h(X,Y) &= d_h(X,Y)/2(H+2). \end{aligned} \tag{3.7}$$

All the denotations have been defined in the previous sections. The reason to choose the square root of the product of the two components instead of the average is because the semantic distance can be shortened by either distance component, not necessarily both. For example, considering an extreme case when the occurrence distance is zero, the semantic distance should be zero too, no matter what the hierarchy distance is. The formal definition of the user-oriented novelty measure is given as follows.

Let a noun phrase NP be a set of keywords, $NP=\{w_1, w_2, \ldots, w_k\}$,

Given an association rule R: A-->C,

Let A be the antecedent, and C be the consequent, where $A=\{NP_1, NP_2, \ldots, NP_m\}$, $C=\{NP_1, NP_2, \ldots, NP_n\}$,

The user-oriented novelty of R is then:

$$un(R) = \frac{1}{mn}\sum_n \sum_m SD(NP_m \in A, NP_n \in C), \text{ and}$$

$$SD(NP_1, NP_2) = \frac{1}{pq} \sum_q \sum_p D(w_p \in NP_1, w_q \in NP_2)$$

$$= \frac{1}{pq} \sum_q \sum_p \sqrt{\left(1 - \frac{P(w_p w_q)}{P(w_p \cup w_q)}\right) \cdot \frac{d(w_p, w_q)}{2(H+2)}},$$

where *un(R)* is the user-oriented novelty of association rule *R*, $SD(NP_1, NP_2)$ *is* the semantic distance between two noun phrases NP_1 and NP_2, and p and q are the number of keywords in NP_1 and NP_2 respectively.

3.3 Summary

This chapter presents the methodology for discovering novel association rules from text. It derives the user's background knowledge from a set of documents provided by the user, and exploits such knowledge in the process of knowledge discovery from text. Keywords are extracted from background documents and clustered into a concept hierarchy that captures the semantic usage of keywords and their relationships in the background documents. Target documents are retrieved from a large corpus by selecting documents that are relevant to the user's background. Association rules among noun phrases extracted from target documents are discovered, and their interestingness is evaluated by a user-oriented novelty measure, which is defined as the semantic distance between the antecedent and the consequent of a rule in the background knowledge.

CHAPTER 4

SYSTEM DESIGN AND IMPLEMENTATION

Chapter 3 presents the user-oriented text mining framework and its application to the discovery of novel association rules from text. This chapter discusses the detailed system design and implementation, including the system architecture, algorithms, and user interfaces. The system is called uMining, which stands for user-oriented text Mining.

4.1 System Design

This section provides an overview of the system architecture and the main processes of the user-oriented text mining system.

4.1.1 System Architecture

Component-based approach is adopted to design the system, uMining. The system includes the following components: *Feature Extractor, Background Knowledge (BK) Developer, Target Document (TD) Retriever, Association Rules (AR) Miner,* and *Rule Evaluator.* These components and their interactions are shown in Figure 4.1

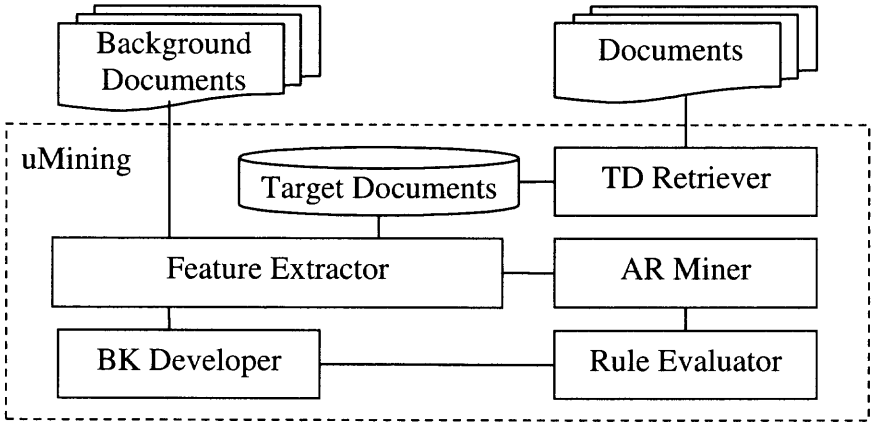


Figure 4.1 uMining system architecture.

The *Feature Extractor* is an important component in the system. It extracts the specified type of features from documents, so that the unstructured documents can be converted to structured vectors of features. Currently it supports two types of features: keyword and noun phrase. When returning the features, it also returns the possible part-of-speech tag(s) associated with each feature.

The *BK Developer* component develops a keyword space that models the user's background knowledge from a list of background documents. First it calls the *Feature Extractor* component to extract keywords from each document. Then the system indexes the extracted keywords and creates an inverted keyword list. It is also capable of developing a concept hierarchy to organize the popular keywords. The inverted keyword list and the concept hierarchy serve as the background knowledge which will be used by other components. This component also provides the functionality for semantic distance calculation (semantic distance is the square root of the product of occurrence distance and hierarchy distance) between two keywords.

The *TD Retriever* component retrieves target documents from a large corpus. The current system supports two retrieval modes: keyword search mode and document similarity mode. Keyword search mode can be used when the content of all documents in the corpus cannot be obtained through direct access. Instead a search interface is provided by the digital library system to find relevant documents and to get the content of documents. A query can be formed from the user's interests or by selecting the most representative keywords from the user's background knowledge keyword list. The query then can be issued to the search interface to find relevant documents. A certain number of documents on the top of the returned document list can be selected as the target

documents. Document similarity mode can be used when the content of all documents in the corpus is available. The user's background keyword list (with frequency information for each word) can be viewed as a virtual document, and the similarity (cosine similarity or Euclidean distance) between the virtual document and each document in the corpus can be calculated. The top N most similar documents can then be selected as the target documents.

The *AR Miner* component discovers association rules among noun phrases from target documents. First, it feeds each target document into the *Feature Extractor* component to extract noun phrases. After converting the target documents into structured vectors of noun phrases, it applies the APRIORI algorithm to discover association rules among noun phrases.

The *Rule Evaluator* component calculates the novelty score of each rule by comparing it with the background knowledge. For each rule, it takes a keyword from the antecedent and another from the consequent, and queries the *BK Developer* component for the semantic distance between the two keywords. After obtaining the distances between all possible combinations, it assigns the average of the distances to the rule as its novelty score.

All components work together to fulfill the task of discovering novel association rules from textual documents. The entire process is not designed to be fully automatic, and the sequence of actions can be triggered by user commands through the user interface. This gives users flexibility when they need to interact with the system during the discovery process. For example, users may want to take a look at the extracted features, and manually remove some useless ones from the list if it is necessary. Users

may also want to try different parameters for such tasks as concept hierarchy development and association rules mining.

4.1.2 System Process Diagram

The system architecture diagram provides an overview of the components and their relationships in the system. Figure 4.2 shows the system process model, which describes the processes and data flows in the system.

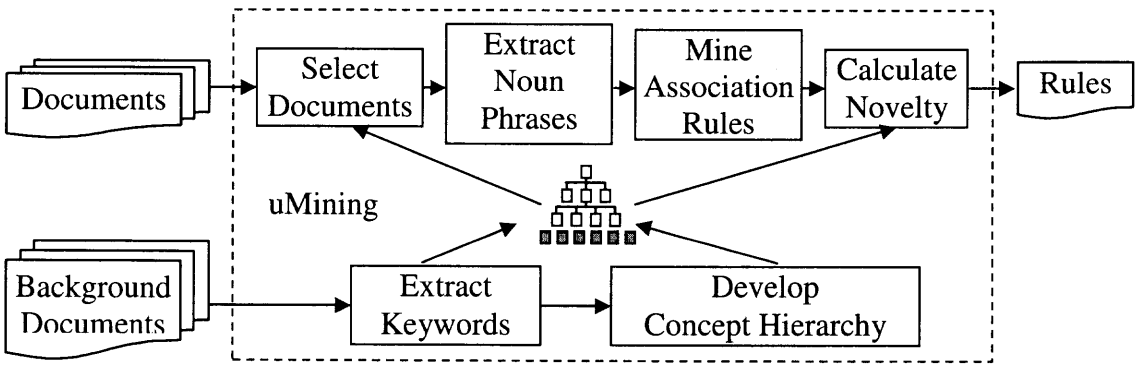


Figure 4.2 System process diagram.

This process model matches the proposed text mining framework in Chapter 3. There are two flows in the system: the knowledge discovery flow and the background knowledge development flow. The knowledge discovery flow includes three stages: pre-processing, mining and post-processing. Pre-processing includes selecting target documents and extracting noun phrases from target documents. During the mining stage, association rules among noun phrases are discovered. Post-processing includes the calculation of the novelty of discovered association rules. The background knowledge development flow includes two processes: extracting keywords from background documents and developing the concept hierarchy.

The two flows are connected through the background knowledge developed by the background knowledge developer. Background knowledge is used to select target documents and evaluate the interestingness of discovered rules.

4.2 Design of Algorithms

There are several algorithms involved in the system design and implementation of uMining. Some of the algorithms are straightforward, while others need to be carefully designed to ensure efficiency and effectiveness. This section discusses the key algorithms in the design of uMining, including the concept hierarchy development algorithm, the hierarchy traversal algorithm, the shortest path search algorithm, and the hierarchy distance calculation algorithm.

4.2.1 Concept Hierarchy Developing Algorithm

The POCA technique (Wu, 2001) states that, for two terms X and Y, if $P(X|Y) > P(Y|X)$, $P(X|Y) >= N$, where $0 < N <= 1$, then X is the parent of Y. A naïve hierarchy development algorithm could build a term-term matrix, calculate the co-occurrence probabilities of all possible term pairs, and determine the parent-child relationship between every two terms. However, because of the high dimensionality of textual data, the number of terms extracted from the background documents can easily reach tens of thousands, which makes the naïve algorithm very inefficient.

Some observations can be made by analyzing the POCA technique. Given X and Y, where $P(X) > P(Y)$,

(1) if X and Y do not satisfy the co-occurrence probability inequality, nor does Y or any child of X.

(2) if X is already an ascendant of Y, there is no need to compare Y and X as well as all children of X again. This situation happens because the POCA technique allows a term (e.g., X) to appear in multiple positions in the hierarchy. When Y is being added to the hierarchy, it is necessary to compare it with X at the first time, but not for the remaining instances of X in the hierarchy.

These two observations indicate that it is not necessary to compare every term pair in order to determine their parent-child relationship, nor is it necessary to compare a new term to all terms already in the hierarchy. A more efficient algorithm is developed as follows.

```
Create an inverted list of keywords in background documents
Let K be the keyword set, and k be a keyword in K
Sort K by document frequency in descending order
Create a virtual keyword r as the root, and set P(r)=1
For Each k in K
  AddKeyWordToHierarchy(r, k)
End For

Function AddKeyWordToHierarchy(h, k)
  If h is already an ascendant of k Then return true
  If P(h|k) > P(k|h) And P(h|k) >= N Then
    Let C be the children of h, and c be a child of h
    For Each c in C
      AddKeyWordToHierarchy(c, k)
    End For
    If k was not added as a child to any c in C
      Add k as a child of h
      Return true
    End If
  End If
  Return false
End Function
```

Figure 4.3 Concept hierarchy developing algorithm.

The algorithm builds the hierarchy incrementally. Each time it tries to add a new keyword to the hierarchy. The correct position of the keyword in the hierarchy is determined by the recursive function *AddKeyWordToHierarchy*. Because the keywords are sorted by their document frequency, any keyword to be added can only become a child of a keyword already in the hierarchy, but not the parent. Also, if *h* and *k* have been already compared, there is no need to repeat the comparison of *k* to other instances of *h* in the hierarchy. These heuristics reduce the comparison times greatly, thus the efficiency of the algorithm is increased.

4.2.2 Depth-first Hierarchy Traversal Algorithm

A hierarchy traversal is needed to search for a path from the root to a node, or from a node to the root. For example, calculating the hierarchy depth H requires the finding of the longest path from the root to a leaf. A depth-first traversal algorithm is outlined as follows.

```
Function DFT(h)
  If h has no child Then
    Return
  Else
    Let C be the children of h
    For Each c in C
      DFT(c)
    End For
  End If
End Function
```

Figure 4.4 Hierarchy depth traversal algorithm.

The searching algorithm performs a depth-first traversal from the root to the leaves of the concept hierarchy. For a specific task, e.g. finding H, additional variables are needed to record the status during the traversal.

4.2.3 Shortest Path Searching Algorithm

This algorithm is designed to find the length of the shortest path connecting two keywords *X* and *Y* in the concept hierarchy. The assumption is that *X* and *Y* are both in the concept hierarchy. Because the concept hierarchy can be viewed as a directed acyclic graph, looking for the shortest path between *X* and *Y* is a classic search problem. Although a depth-first or best-first search can find the optimum solution, they perform badly when the number of nodes is huge, as is true in this case where the number of the unique keywords in background documents is about tens of thousands.

```
Function ShortestPathLength(X, Y)
  Depth-first search the shortest path from X to r, P_Xr
  Depth-first search the shortest path from Y to r, P_Yr
  Remove the successive common nodes from the right side of P_Xr and P_Xr,
    record the new paths as P_Xr' and P_Xr'
  Set upper = length of P_Xr' + length of P_Xr'
  Let P be the current searching path (initially empty)
  Append X to an empty queue Q
  If Q is empty Or searching time is greater than n seconds Then
    return upper
  End If
  Remove node N from the front of Q
  Append node N to the tail of P
  If length of P is not less than upper Then
    Remove N from the tail of P
    Repeat from line 8
  End If
  If N = Y Then
    upper = length of P
    Remove N from P
  Else
    Append the children and parents of N that are not in Q to the end of Q
  End If
  Repeat from line 8
End Function
```

Figure 4.5 Shortest path searching algorithm.

A branch and bound search can greatly speed up the search by taking consideration of the existing knowledge learned as far. If a partial solution cannot improve on the best, it is abandoned. The search algorithm is outlined in Figure 4.5.

The branch and bound search requires an upper bound which represents the best solution found so far, and a lower bound which represents the cost of the current action. The upper bound is initially set to the length of a path connecting *X* and *Y* found by locating the path between *X* and the root and the path between *Y* and the root, and removing the common nodes, if any, in both paths at the root side (See Figure 4.6 for an example). The lower bound is the length of the current search path. If the lower bound is greater than the upper bound, the search is abandoned because it will not find any shorter path than the current shortest path.

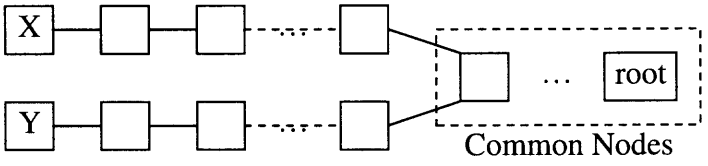


Figure 4.6 Locating a path between *X* and *Y* through the root.

A queue is maintained to store all keywords that need to be appended to the current search path. If introducing a new keyword makes the current path longer than or equal to the shortest path found so far, this search path is abandoned because no better solution will be found by following the search path. Therefore, the new keyword is removed from the tail of the current search path, and the search continues by expanding the current path with other keywords in the queue.

When the queue is empty, all possible paths have been searched, and the length of the shortest one is saved in the variable *upper*. This is the ideal ending condition for the

search. However, the expansion of the search path may result in a combinatorial explosion, which may lead to an unreasonable time to find the optimal path. A time-limit n is set to avoid this problem. If within the time limit the search function has not gone through all possible search paths, it returns the length of the current shortest path.

4.2.4 Hierarchy Distance Calculating Algorithm

The novelty of an association rule is defined as the semantic distance between the antecedent and the consequent of the rule. It is further defined as the average of the semantic distances between the keywords in the antecedent and the keywords in the consequent. Calculation of the distance between two keywords is denoted in Formula 3.6. The inverted list of the keywords in background documents provides a convenient way to calculate *P(X)*, *P(Y)*, *P(XY)* and *P(XUY)*, which can be used directly to calculate the occurrence distance between X and Y. Calculating the hierarchy distance between X and Y, d(X, Y), is more difficult. The algorithm for calculating d(X, Y) is outlined as follows.

```
Let K be the background knowledge keyword space
Let CH be the concept hierarchy
Let H be the depth of the concept hierarchy
Let X and Y be the two keywords

If X∉ K and Y∉ K Then
   d(X, Y) = 2*(H+2)
Else If X∈ (K-CH) and Y∉ K Then
   d(X, Y) = (H+1) + (H+2)
Else If X∉ K and Y∈ (K-CH) Then
   d(X, Y) = (H+2) + (H+1)
Else If X∈ CH and Y∉ K Then
   d(X, Y) = ShortestPathLength(X, r) + (H+2)
Else If X∉ K and Y∈ CH Then
   d(X, Y) = (H+2) + ShortestPathLength (Y, r)
Else If X∈ (K-CH) and Y∈ (K-CH) Then
   d(X, Y) = (H+1) + (H+1)
Else If X∈ CH and Y∈ (K-CH) Then
   d(X, Y) = ShortestPathLength (X, r) + (H+1)
Else If X∈ (K-CH) and Y∈ CH Then
   d(X, Y) = (H+1) + ShortestPathLength (Y, r)
Else If X∈ K and Y∈ CH Then
   d(X, Y) = ShortestPathLength (X, Y)
End If
```

Figure 4.7 Hierarchy distance d(X, Y) calculating algorithm.

The hierarchy distance calculating algorithm checks the locations of *X* and *Y* in the background knowledge, and calculates the distance score *d(X, Y)* according to the formulas summarized in Table 3.1.

The key algorithms have been implemented in the current system. The VC++ source code is attached in Appendix G.

4.3 System Implementation

The system, uMining, is implemented in Microsoft Visual C++. This section describes the system user interface and some key functions.

4.3.1 Main User Interface

The main user interface is shown in Figure 4.8.

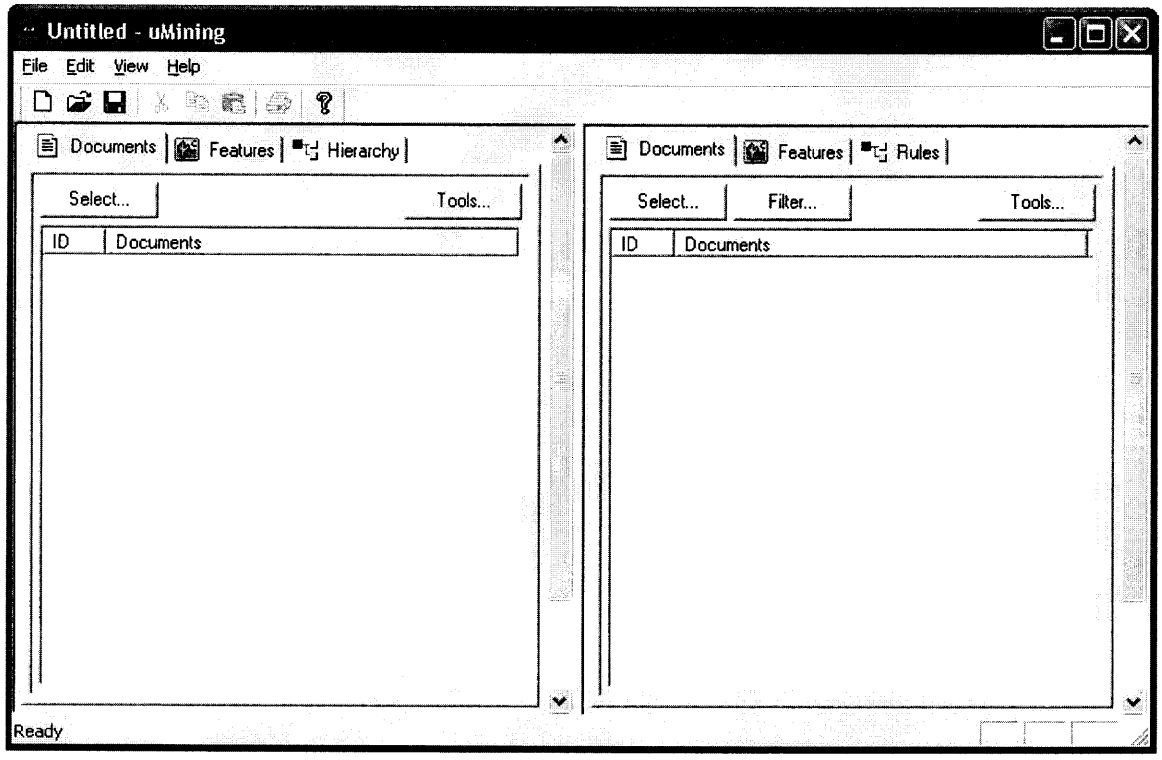


Figure 4.8 uMining main user interface.

The main window is divided into two areas: the working area for background knowledge development on the left and the working area for knowledge discovery on the right. In each working area, there are a few tab windows, and each of them is implemented for a certain step/task of the entire text mining process.

4.3.2 Background Knowledge Development Window

The background knowledge development area has three tab windows: Documents, Features, and Hierarchy, which correspond to background document selection, feature extraction from background documents, and concept hierarchy development, respectively.

4.3.2.1 Background Document Selection. When the *Select* button in the document tab is clicked, a document selection dialog will pop up, as shown in Figure 4.9.

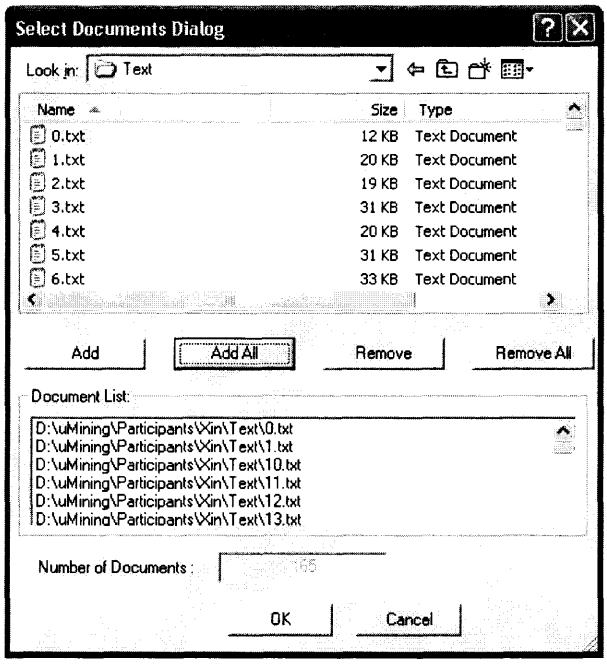


Figure 4.9 Background documents selection dialog.

In this dialog, the user first locates the directory where the background documents are stored, and then adds the documents to the background document list.

4.3.2.2 Feature Extraction from Background Documents. After selecting the background documents, the user can switch to the *Features* tab. The feature style for background knowledge development is *Word*. The *Command* button triggers the pop up of a command menu. Clicking the *Extract Features* from the menu will start the process

of extracting keywords from the background documents. The extracted keywords are displayed in a list with their total frequency, document frequency, and possible part-of-speech tags. The list can be sorted by one of the attributes in the headers when the corresponding column is clicked.

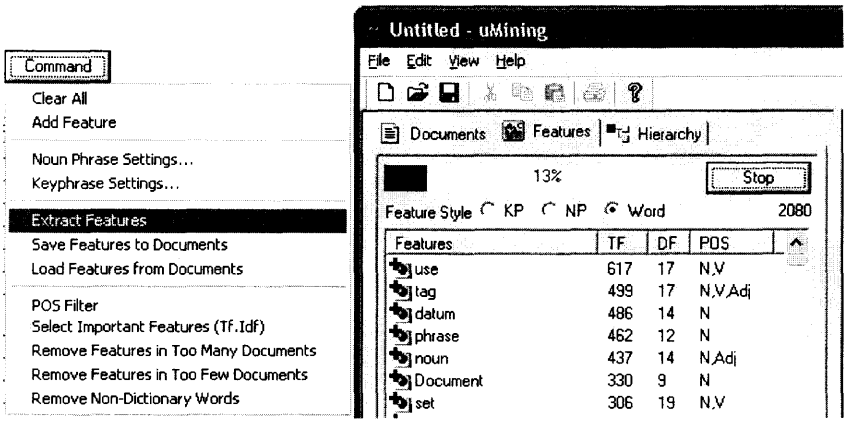


Figure 4.10 Feature (keywords) extraction from background documents.

After keywords are extracted, the user can further refine the features by the commands in the last group of the pop up menu. For example, the user may want to apply the TF.IDF measure to select important keywords from each document, or simply remove keywords that appear in too many documents. For either action, the user can specify a threshold to tell the system what features should be kept or removed.

4.3.2.3 Hierarchy Development. The last step of background knowledge development is the concept hierarchy development, as shown in Figure 4.11.

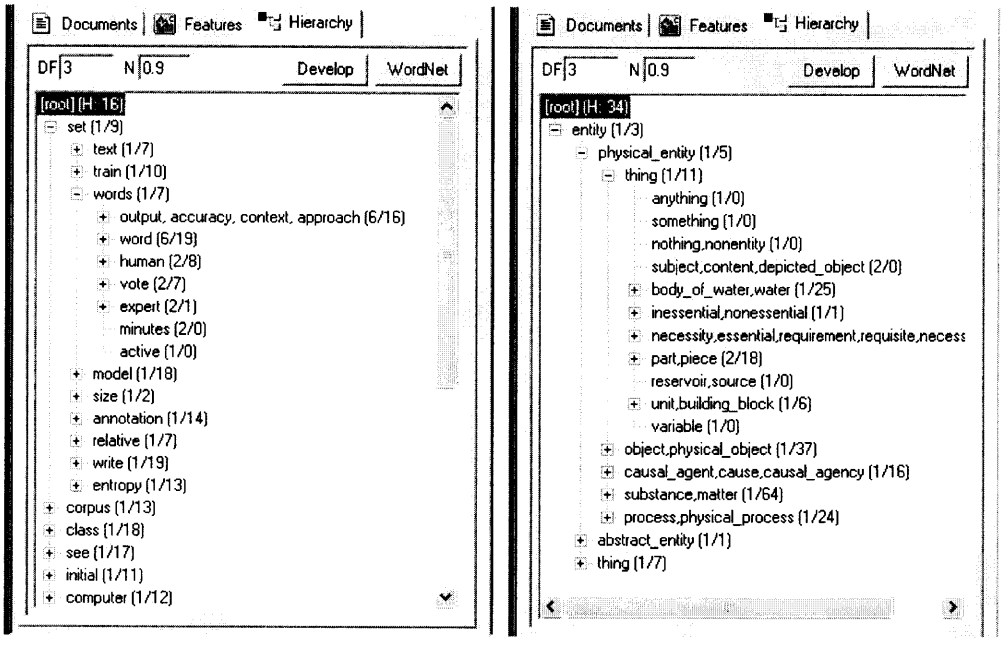


Figure 4.11 Concept hierarchy development (POCA vs. WordNet).

Two types of concept hierarchy can be developed: through the POCA technique and through the WordNet lexical database. For the POCA technique, there are two parameters that need to be configured, the document frequency (DF) and the co-occurrence probability threshold N. For the WordNet concept hierarchy, the WordNet lexical database is needed. The concept hierarchy is developed by extracting concepts from the WordNet lexical database and linking concepts by following the pointers from one synset (a synonym set; a set of words that are interchangeable in some context) to others in WordNet. The WordNet concept hierarchy can be used as general knowledge to evaluate the interestingness of association rules (Basu, 2001).

4.3.3 Knowledge Discovery Window

The knowledge discovery area has three tab windows: Documents, Features, and Rules, which correspond to target document selection, feature extraction from target documents and association rules mining and evaluation respectively.

4.3.3.1 Target Document Selection. The system provides two utilities for target document preparation (downloading files and converting them to text files). The retrieval utility retrieves the target documents from a search system, and the converter utility converts document files of some types (.doc, .pdf, .htm) to plain text format. An example of searching and downloading documents from Google Scholar is shown in Figure 4.12. The user can specify the query as well as the number of documents to be retrieved. Once the target documents are saved on the computer in plain text files, they can be added to the target document list in the same way the background documents are selected.

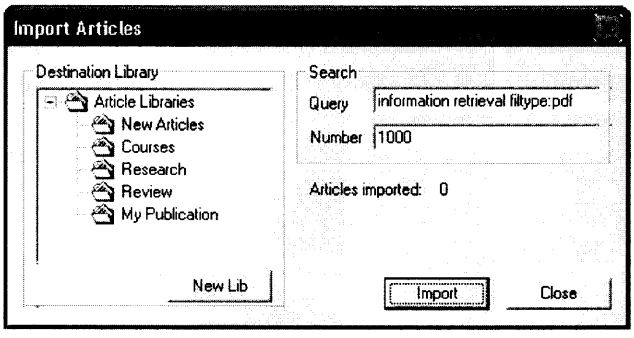


Figure 4.12 Retrieve target documents from Google Scholar.

The system also provides users with another option to select target documents if the user has a large document collection on hand already and wants to select documents that are relevant to his/her background or interests. In this case, the user first needs to select background documents and extract keywords (features) from background documents. The keyword list is used to develop a virtual document, and the cosine

similarity between the virtual document and each document in the large collection is computed. A similarity threshold can be specified to select a certain number of similar documents. The process is called filtering (see the *Filter* button in Figure 4.8).

4.3.3.2 Feature Extraction from Target Documents. Noun phrases are extracted from target documents. The noun phrase setting dialog allows the user to specify the noun phrase pattern and the minimum number of words in the extracted noun phrases. An example is shown in Figure 4.13.

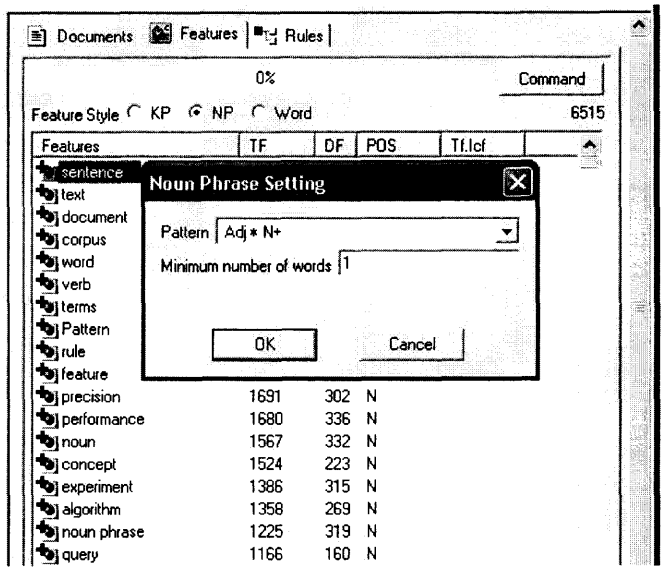


Figure 4.13 Noun phrase extraction from target documents.

The extracted noun phrases, as well as their term frequency (TF) and document frequency (DF) are displayed in the list window.

4.3.3.3 Association Rules Mining and Evaluation. The APRIORI algorithm is implemented to discover the associations among noun phrases extracted from target documents. The user can specify the support and the confidence threshold, and trigger the discovery process by clicking *Generate* from the pop up menu associated with the *Command* button. The *Calculate Novelty* command will call the appropriate functions to

update the novelty scores of the selected rules, or all the rules if none is selected. The novelty scores are normalized to 1 (the least) to 7 (the most).

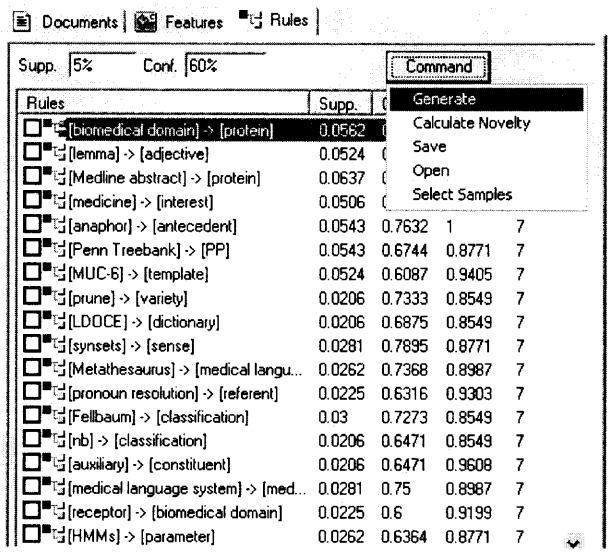


Figure 4.14 Association rules mining and evaluation.

Figure 4.14 shows some rules as well as their support, confidence, and novelty scores. The results can also be sorted by one of the three measures by simply clicking the corresponding column header. The list can be exported to a text file, which can be further imported into Microsoft Excel or other statistical software for further analysis.

4.4 Summary

This chapter discusses the system design and implementation. The design includes system architecture design, system process design, and algorithm design. The system, uMining, is implemented as a GUI application in VC++. Important operations and the corresponding user interfaces are illustrated with screen shots.

CHAPTER 5

USER EVALUATION

In the literature review, challenges of current text mining systems and limitations of existing interestingness measures are identified. This study aims at developing an interestingness measure to evaluate the novelty of discovered association rules without asking for explicit users' expectations. This goal is achieved by implicitly deriving users' background knowledge from a set of documents and utilizing the background knowledge to evaluate the novelty of association rules. A novel rule reflects something absent and/or a relationship deviating greatly from the normal pattern in the user's background knowledge.

The user-oriented novelty measure is design to help users find previously unknown and potentially useful rules from a large amount of rules discovered by the text mining system. However, it remains unclear (1) whether the novel rules identified by the system are really novel to the user, (2) whether novel rules are also useful, and (3) what user factors can affect the performance of the novelty measure in terms of identifying novel and useful rules. A user study was conducted to answer these questions. In the study, 25 PhD students submitted their research articles they have read as the background documents, and evaluated the novelty and usefulness of the association rules that were discovered from the target documents retrieved online. The proposed user-oriented novelty measure was compared with the users' subjective novelty and usefulness ratings to evaluate its performance, and it was also compared with other interestingness measures for their performance in identifying interesting and useful associations.

This chapter first presents the research questions. Then, it briefly introduces the pilot study conducted to preliminarily investigate the feasibility of the proposed method. Last, it describes the design of the formal study, including participant invitation, the user study procedure, the evaluation system and the instruments.

5.1 Research Questions

There are three major concerns about the performance of the proposed user-oriented novelty measure in the evaluation: (1) accuracy of novelty prediction (novelty scores calculated by the algorithm vs. user subjective novelty ratings), (2) usefulness indication power (novelty scores calculated by the algorithm vs. user subjective usefulness ratings), and (3) effects of user factors (e.g. gender, experiences, and so on). These concerns lead to the research questions listed as follows.

Q1. Can the user-oriented novelty measure help users find interesting rules?

The proposed method predicts whether a rule is interesting by assigning it a novelty score. Since the goal of a text mining system is to find previously unknown and potentially useful knowledge from documents, novelty and usefulness can be two aspects for interestingness evaluation. Given the two evaluation aspects, research question Q1 can be further decomposed into the following two sub-questions.

Q1.1. Is the novelty prediction accurate?

This question examines whether rules ranked as novel by the user-oriented novelty measure are really novel to the user. If rules having higher user-oriented novelty scores are also rated more novel by the user, the prediction is accurate. Finding the

answer to this question requires a comparison of the novelty predictions calculated by the system with the actual novelty scores rated by the user.

Q1.2. Are novel association rules also useful to users?

A rule could be interesting if it is novel, but it becomes beneficial if it is also useful. Novelty measures can be used to identify useful rules only if there is a correlation between novelty ratings and usefulness ratings by the users. This research question will examine whether there is a close relationship between user novelty ratings and user usefulness ratings of association rules. If the relationship can be established, further analysis can be done to investigate how well the user-oriented novelty measure can identify useful rules. The analysis can be done by comparing the novelty predictions calculated by the system with the actual usefulness scores rated by the user.

Q2. Does the user-oriented novelty measure perform better than other interestingness measures?

There have been different types of measures proposed to evaluate the interestingness of association rules. Based on the classification of existing interestingness measures, this question can be investigated with the following sub-questions.

Q2.1. Does the user-oriented novelty measure perform better than other novelty measures in predicting the novelty of association rules?

Other novelty measures, such as the WordNet novelty measure, also aim at predicting the novelty of association rules. This question will examine which novelty measures are more correlated with the subjective novelty ratings by users. Other types of

interestingness measures, such as objective measures, will not be compared, because they are not designed to predict the novelty of association rules.

Q2.2. Does the user-oriented novelty measure perform better than other interestingness measures in terms of identifying useful association rules?

Because usefulness of association rules cannot be directly predicted, each interestingness measure emphasizes a particular aspect. For example, novelty measures focus on the unknownness of association rules. Though different measures are measuring different aspects of association rules, finding useful rules is one of the common goals. Therefore, it is fair to compare the performance of all interestingness measures in terms of identifying useful association rules. The comparison will examine which interestingness measures are more correlated with the subjective usefulness ratings by users.

Q3. What are the factors affecting the performance of the user-oriented novelty measure?

The novelty calculation process includes several steps, and during each step there are various parameters involved. In addition, users can be different. Finding the major affecting factors will help understand how the system works, what is needed to get better results, and what types of users the system is more oriented to. The evaluation will investigate two major factors: the number of background documents and user factors.

Q3.1. How many background documents are needed?

Background documents are very important for the system to develop the background knowledge, which will further affect the novelty calculation. In order for the background knowledge to represent what a user knows, a sufficient number of

background documents are required to cover the user's knowledge as much as possible. On the other hand, if too many background documents are required for the system to produce accurate results, users may become reluctant to use the system. This research question will investigate the relationship between the system performance and the number of background documents, so that the minimum number of background documents required to run the system can be detected.

Q3.2. Does user difference have effects on the system performance?

The proposed method does not limit itself to a specific type of users. This question investigates whether the difference in users causes difference in the results.

5.2 The Pilot Study

A pilot study was conducted to preliminarily test the feasibility of the proposed approach. The pilot study was mainly concerned with the prediction accuracy of the user-oriented novelty measure (research question 1).

5.2.1 Pilot Study Design

The proposed approach was applied to a real text mining task: discovering unknown knowledge from competitors' websites. The web pages on the website of the Information Systems (IS) Department at New Jersey Institute of Technology (NJIT) were taken as the background documents, and the web pages on competitors' websites were considered target documents. The goal was to find novel association rules from the target web pages.

Competitors' websites were chosen by issuing a query "information systems department site:edu" to the Google search engine, and the returned top ten IS related

departments were selected. The websphinx crawler (Miller and Bharat, 1998) was used to download all the web pages from each website. Documents in HTML, DOC, and ASP format were converted to plain text files, while files in other formats were excluded from processing.

After conversion, 512 background documents and 1,422 target documents were obtained. From the background documents, 12,945 keywords were extracted. A minimum document frequency of 6 and a probability threshold of 0.8 were chosen for hierarchy development. Such settings resulted in a hierarchy whose depth was 13. Noun phrases were extracted from target documents, and the TF.IDF weighting scheme was applied to remove the low 20% noun phrases from each target document. The support and confidence constraints were set to 1% and 60% respectively for association rules mining. A low support was chosen to increase the recall of useful rules. In this pilot study, only two-item rules were generated, but it is not difficult to generate rules with more noun phrases. Novelty was calculated for each rule and was normalized from 1 (the least) to 5 (the most).

Eight subjects from the IS department at NJIT were invited to evaluate the discovered association rules. Considering the large number of discovered association rules, it is not feasible to ask participants to evaluate all rules. With respect to previous research, Basu et al. (2001) selected 100 rules (25 rules for each of the 4 groups) from the final rule set for evaluation. Similarly, 80 rules (16 rules at each novelty level) were randomly selected from the discovered rules in this study. A web interface was created for the participants to rate the novelty of the sample rules online. The subjects were first asked to spend 30 to 50 minutes browsing NJIT IS department's website. The purpose

was to reinforce the background knowledge during the evaluation. The novelty of a rule as well as its usefulness was evaluated at a 5-point Likert scale (1 for the worst, and 5 for the best). Participants were asked to choose a novelty level of a rule according to the knowledge they obtained from the NJIT IS department website.

5.2.2 Results

A total of 4,922 association rules were discovered. Table 5.1 shows the breakdowns of the number of rules at different novelty levels. It shows that nearly 80% of the rules have a novelty value less than 5. Table 5.2 shows some selected rules with their support, confidence, and novelty scores.

Table 5.1 Breakdowns of the Number of Rules

Novelty	Number of rules
5	33
4	52
3	796
2	271
1	3,770

Table 5.2 Some Selected Rules

Rule	Supp	Conf	Novelty
[Auditor] -> [MSBA]	0.09	1	5
[Trademarks] -> [Karl Eller Center]	0.09	0.8	4
[Admission] -> [Assistantships]	0.07	0.8	3
[Final Exam] -> [Office Hours]	0.09	0.8	2
[Phone] -> [Office]	0.10	0.7	1

The intersubject agreement is investigated to ensure that a rule is not judged more novel than others for any reason other than chance. Two statistical techniques, the Kappa Statistic K and the Kendall's Coefficient of Concordance W were calculated to ensure the level of intersubject agreement. Table 5.3 shows the results of K testing. Overall the

result is significant at 0.005 level, so the hypothesis that the agreement occurs merely by chance is rejected.

Table 5.3 Kappa Statistics K for Nominal Response

Novelty	K	Std. Err.	z	Prob>Z
1	0.073	0.028	2.63	0.004
2	0.001	0.028	0.05	0.482
3	-0.040	0.028	-1.46	0.928
4	0.101	0.028	3.62	0.0001
5	0.129	0.028	4.62	<0.0001
Overall	0.050	0.015	3.31	0.005

Table 5.4 Kendall's Coefficient of Concordance W

W	F	Denom. Num. DF	DF	Prob>F
0.432	3.809	2.63	423.33	<0.001

However, K is not an optimal measure because it considers agreement on unordered categories. It could happen that one subject always rates a rule one point lower than another subject, so the agreement between them is 0, though they agree on the order of the rules. The Kendall's Coefficient of Concordance W was used to avoid the problem by measuring the agreement between subject's relative rankings of rules. The results are shown in Table 5.4. The Kendall's Coefficient of Concordance W demonstrates that there are significant levels of agreement between subjects' assessment. It is concluded that the intersubject agreement is sufficient for further investigation into correlation between human judgments and system predictions.

5.2.3 Correlations

Both the Pearson's raw score correlation and the Spearman's rank correlation were calculated, as shown in Table 5.5.

Table 5.5 Correlations

	Human-Human		Human-Algorithm	
	Raw	Rank	Raw	Rank
Mean	0.315	0.270	0.444	0.415

The results show that the correlation between the system and the human subjects is comparable to, or even slightly better than, that between the human subjects. The significance t-tests are all above the minimum significant r at the $p<0.01$ level of significance. This leads to the conclusion that the correlations are not due to the random chance. The results of the correlation tests indicate that the system accurately predicts the novelty of association rules.

5.2.3 Discussion

The pilot study shows that the algorithm is effective in predicting the novelty of association rules, as evidenced by the fact that the correlation between human judges and the algorithm is comparable to the correlation among human judges. Using the novelty measure to rank the rules can eliminate about 80% of the uninteresting rules if a mid-point novelty value is selected.

5.3 Formal User Study Design

The pilot study is a preliminary study which investigates only the novelty prediction accuracy. A formal user study was conducted to seek answers to all of the research questions.

5.3.1 Overview

The proposed method can be applied to the task of mining novel association rules from different types of documents. In the formal user study, the problem was defined as mining novel association rules from research papers for academic researchers.

PhD students in the Information Systems domain were invited to participate in this study. Participants first submitted their research articles (background documents) through one of the article-uploading tools provided by the study. The investigator then ran the text mining system, uMining, to discover novel association rules for each participant, using the information and the articles provided by each participant. A sample set of rules was created for evaluation from the discovered association rules using a stratified sampling method. Last, participants were asked to evaluate the sample rules, and complete a post-evaluation questionnaire.

The evaluation process consists of the following steps:

- Register online
- Install the tool
- Upload articles
- Discover rules
- Evaluate rules
- Complete questionnaire

Figure 5.1 User study status.

A website and a software tool were developed to support the evaluation process (http://highlight.njit.edu/uMining/). On the website, after a user logs in, the system shows the current status of the evaluation process. The completed steps are marked with a blue mark (see Figure 5.1).

5.3.2 Participants

Invitation letters were sent to PhD students in Information Systems departments through different channels. The PhD students in the IS Department at NJIT were invited directly by emails and by a message posting in the community electronic bulleting system. The invitation letter was also sent to the contacts of the IS related departments at other universities (information was collected from the AISNet.org website), and about half of letters were forwarded to the PhD students in those departments.

PhD students who were willing to participate in this study were provided with detailed instructions. First, they needed to create an account at the user study website. During the registration, participants were asked to sign the consent form (see Appendix B). They also needed to answer a background questionnaire which asked for some basic demographic information and research interests of the participants (see Appendix C).

The screen shots for the consent form and background questionnaire are shown in Figure 5.2. The consent form and the background questionnaire can be found in Appendix B and C.

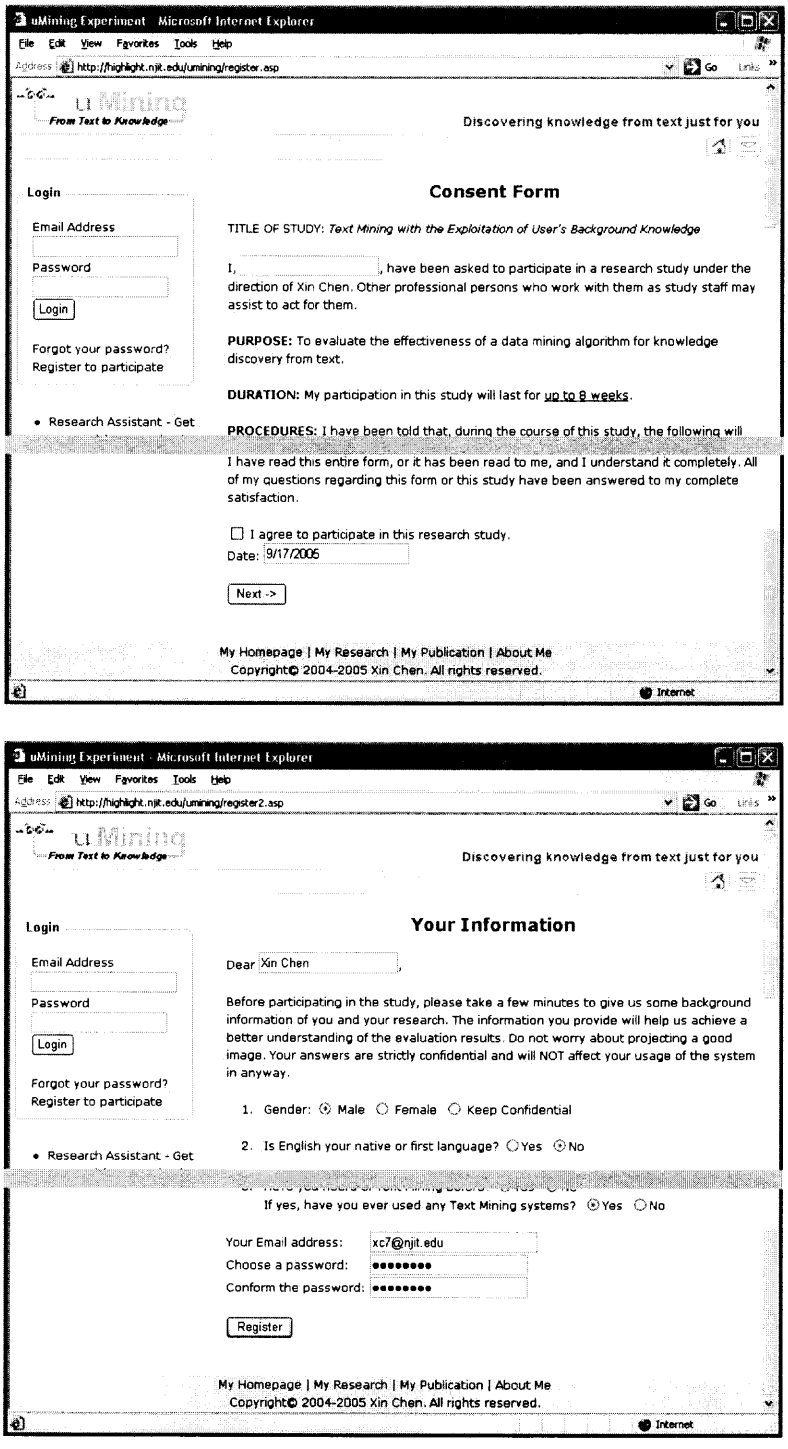


Figure 5.2 Consent form (up) and the background questionnaire (bottom).

In total there were 35 people registered in the system, and 25 of them completed the entire user study. Table 5.6 provides a summary of the information of the 25 participants.

Table 5.6 Summary of Participant Information

Stage	Number	Research Interests
Looking for a topic	3	Information Retrieval, Text Mining, Knowledge Management, DSS, GDSS, HCI, Information Security, Recommendation Systems, Health Informatics, Visualization, Distance Learning, Collaborative Learning, System Modeling, Emergency Management, E-commerce, Merge & Acquisitions
Completing SOTA*	3	
Completing proposal	9	
Completing dissertation	8	
Finished PhD study	2	

* SOTA: State-of-the-Art literature review

Most of the participants are at the stage of completing their proposal or dissertation. This means that they have a clear research topic in mind and have been reading research articles related to their research. Participants have a variety of research interests, as shown in Table 5.6. The diversity in research interests provides a good test bed for investigating whether the proposed methodology works for people with different backgrounds.

5.3.3 Background Document Collecting

After the participants registered online, they were provided with detailed instructions on how to submit their background documents. One of the instructions was to specify what they should submit. Participants were informed that only the articles that were used in the preparation for their research were needed. When submitting an article, participants needed to choose where the articles was referenced, such as in their SOTA literature review, proposal, dissertation, and publications. To give participants more flexibility, the

investigator provided two different ways for article submission: through an article management tool and through the online user study system.

5.3.3.1 The Article Management Tool. A tool, named *Research Assistant*, was developed for managing research articles. Participants were informed to download the tool from the user study website, and install it on their own computers. *Research Assistant* organizes research articles by grouping them into different categories, each of which could correspond to one research area of the participant or one type of work where the included articles were referenced. The main user interface is shown in Figure 5.3.

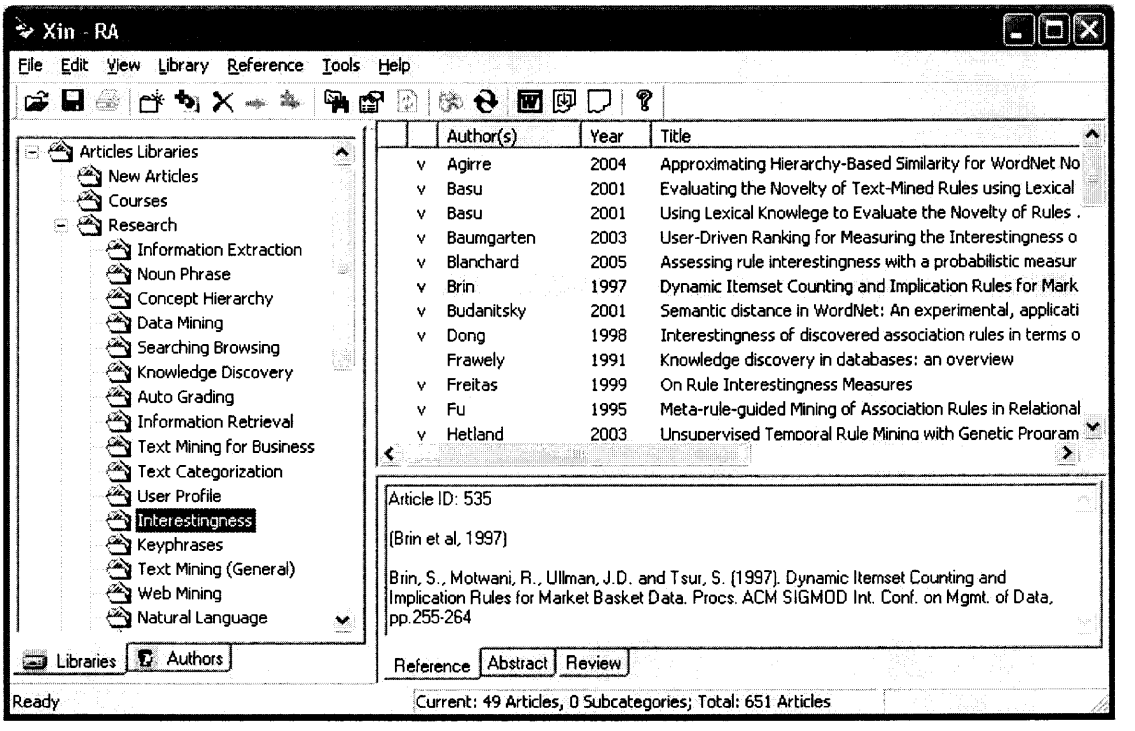


Figure 5.3 User interface of *Research Assistant.*

Users can create hierarchical categories in the left window, and add articles to each category. They can select a category to display the contained articles in the list window. When an article is selected, the full reference is shown in the bottom window.

When adding an article, the user needs to specify the article information and other related information. The interface is shown in Figure 5.4.

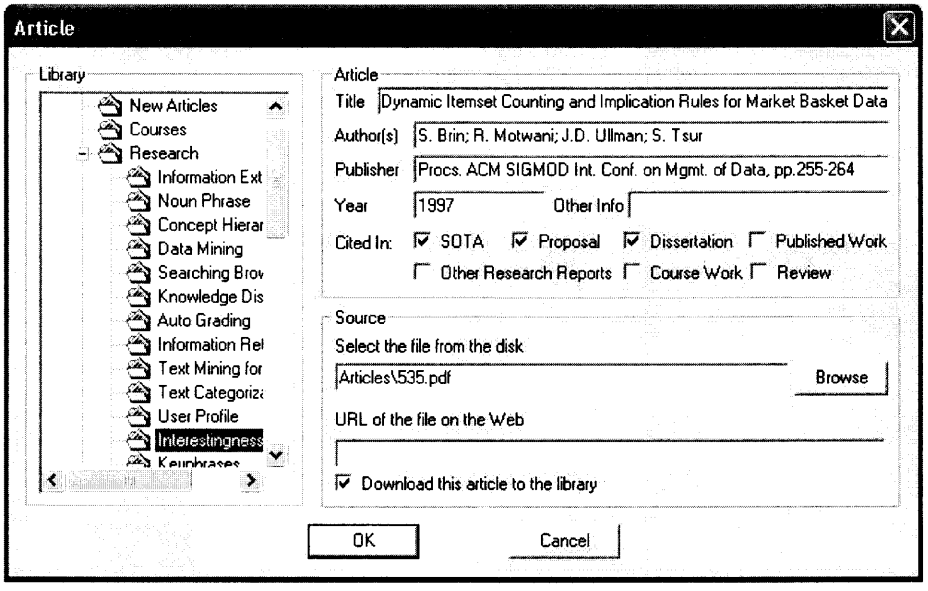


Figure 5.4 Add an article to a category.

After inputting the basic information of an article, the user can specify where the article was referenced by checking the appropriate checkboxes. The user can also link this article to a file on the hard disk and/or a URL on the web. When only a URL is provided, the system will try to download the file from the Internet. *Research Assistant* provides a more convenient way to obtain the required information when it is available in other sources, such as a Microsoft Word document or an Html webpage. The user can select a reference text block in Word or Internet Explorer, and click a button to send the text to *Research Assistant*. The *Reference Parser* in *Research Assistant* will parse the reference block and fill in the fields in the dialog automatically. Importing functions are also available to import references from EndNote®. Once the articles are added to *Research Assistant*, they can be uploaded to the server hosting the user study website automatically (see Figure 5.5).

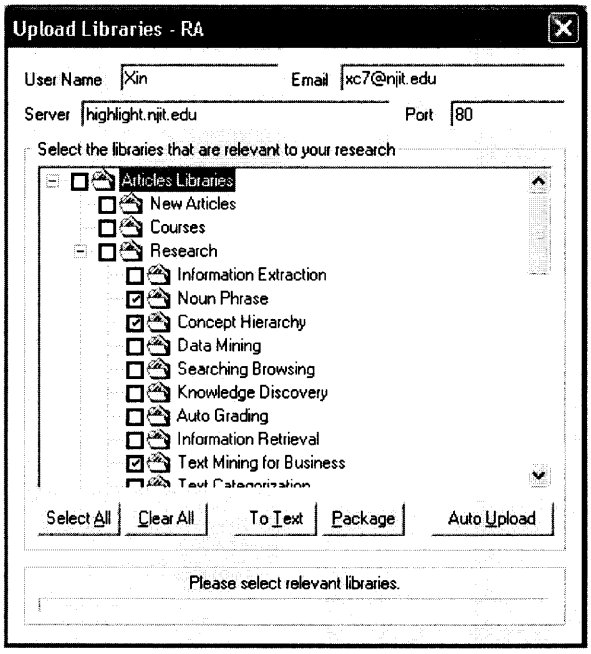


Figure 5.5 Upload articles from *Research Assistant.*

The user first needs to provide the login information (the registered user name and email), and then select the categories that are related to his/her research. Because the tool is designed for managing all articles, there could be categories created for other purposes (e.g. course work). Only the articles that are related to the participant's research are needed. The uploading process is started by clicking on the *Auto Upload* button. The uploading process includes creating a package containing all the articles in the selected categories, connecting to the server, and uploading the package to the server. A notification email is sent to the investigator after each successful upload.

5.3.3.2 Online Submission. The upload tool was implemented on the Windows platform. Some participants were using computers with other platforms (e.g. UNIX and Macintosh). An online uploading function was implemented for those participants. Because the online system is purely HTTP-based, it can be accessed by all computers with a web browser installed. After a user logs into the online system, he/she can go to

his/her article libraries and click *Add an Article* to open the article submission page, as shown in figure 5.6.

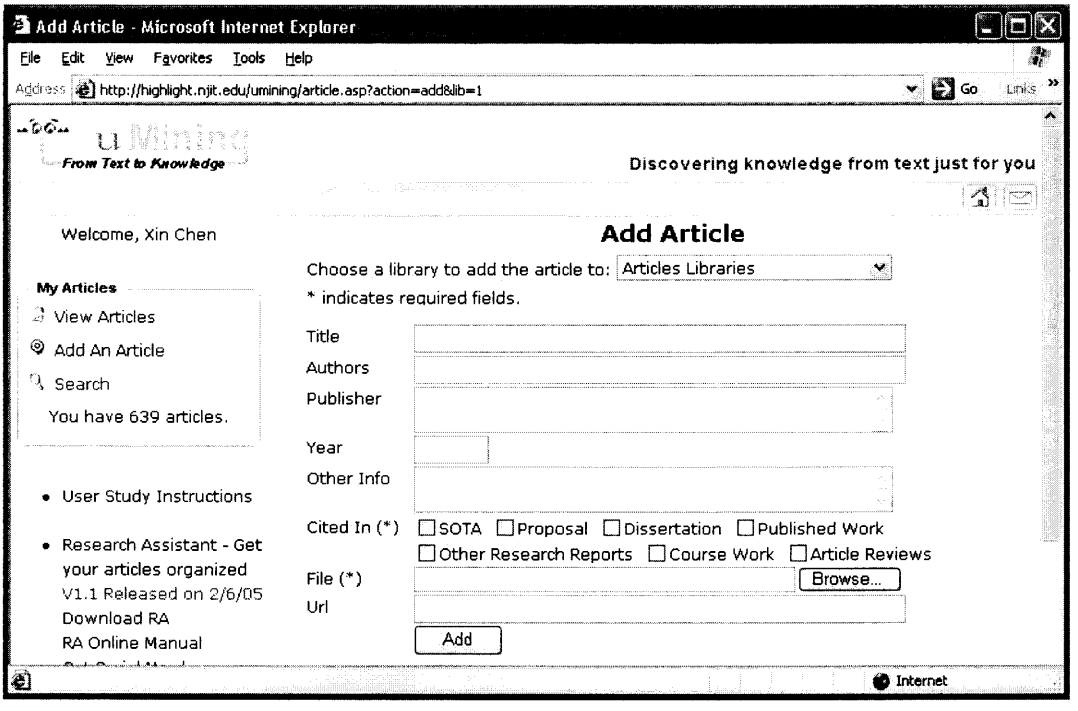


Figure 5.6 Online uploading an article.

The fields in the upload page are similar to those in *Research Assistant*. The uploaded articles are saved in the directory created for the participant. When the participant chooses to upload articles online, he/she needs to notify the investigator by email once he/she has finished the uploading.

5.3.3.3 Summary of Background Documents. Because it is impossible to set up a fix number of articles that should be submitted for all participants, the number of background documents submitted is different from one participant to another. Table 5.7 shows the ranges of the number of background documents and the corresponding number of participants.

Table 5.7 Summary of Background Documents

Number of background documents	Number of participants
<20	2
20 ~ 29	4
30 ~ 39	7
40 ~ 49	4
50 ~ 59	4
>= 60	4

The two different submission methods are provided to participants for their convenience. When a participant has chosen what articles to submit and uploaded them to the online system, the documents will be stored and processed in the same way. Therefore, the uploading tool will have no effect on the results the investigator wishes to investigate.

5.3.4 Knowledge Discovery

After a participant registered online with his/her research interests, the investigator ran the retrieval utility to build a collection of target documents for the participant; after he/she provided the background documents, the investigator ran the uMining program to discover association rules from the target documents and calculate the novelty score of each rule. The process has been described in chapter 4, and this section will discuss the system setup (e.g. the parameter values) used in the user study.

5.3.4.1 Background Knowledge Development. Participants were informed to submit their background documents in PDF, DOC, and HTML format, and these files were converted to plain text files. Keywords were extracted from each background document, and a concept hierarchy was developed. The document frequency constraint (DF) and the co-occurrence probability (N) were set to 20% and 0.8 respectively.

5.3.4.2 Target Document Retrieval. For each participant, a query was formulated based on his/her research interests, and issued to the Google Scholar (scholar.google.com) search engine to retrieve relevant articles. Google Scholar is a search engine designed specifically for searching scholarly literature, including peer-reviewed papers, theses, books, preprints, abstracts and technical reports from all areas of research. Articles from these sources would be of great interests to a participant, if they match his/her research interests.

Because by default Google Scholar also returns links to offline publications, such as articles in digital libraries and references in books, and it is difficult to obtain the content of these publications, the target document retrieval was limited to online resources only. It was done by specifying the *filetype* parameter in each query to retrieve only PDF files. For example, if a participant's research interest is *information retrieval*, the query used to retrieve target documents will be *"information retrieval" filetype:pdf.* If a participant specified two interests, the retrieval process was repeated twice with one interest at a time. Similar process was done for participants with more than two interests. A total of 1,000 documents were downloaded. If a participant had more than one research interests, the total number was evenly divided by the number of queries formulated from his/her interests for each retrieval iteration.

The downloaded PDF files were converted to plain text files. Not all PDF files could be converted, because some of them were invalid PDF files (e.g. an error message was returned when the URL returned by Google Scholar was no longer valid), and some of them did not contain text (e.g. PDF files created from scanned images). Table 5.8

shows a summary of the converted target documents. Most of the participants have about 300 to 600 target documents.

Table 5.8 Summary of Target Documents

Number of target documents	Number of participants
<300	2
300 ~ 399	6
400 ~ 499	5
500 ~ 599	9
>= 600	3

5.3.4.3 Feature Extracting. Noun phrases were extracted from each target document for every participant. The part-of-speech tagger used in this study was a revised version of the widely used Brill tagger (Brill, 1992; Brill, 1995), and the part-of-speech pattern of noun phrases was defined as A* N+, where A refers to adjective, N refers to noun, * means none or more instances, and + means one or more instances. The minimum length of noun phrases was set to 1, which meant there was no restriction on the number of words in a noun phrase. The TF.IDF scores were assigned to noun phrases as their weights. From each target document, the low 20% noun phrases were removed, and the remaining noun phrases were saved as its document features.

5.3.4.4 Association Rules Mining and Novelty Calculation. The APRIORI algorithm was implemented to identify association rules between groups of noun phrases. *Support* and *confidence* were set to 2% and 60% respectively. Novelty was calculated for each rule, and normalized from 1 to 7. Because the number of discovered rules for each participant was more than 20 thousand, it was not feasible to ask the participant to evaluate them all. A stratified sampling method was used to select 9 rules from each novelty level, and in total a sample of 63 rules was created for each participant. The

reason to choose 63 rules is that the participant can finish the evaluation within one and a half hours. After the sample set was created, the rules in the set were shuffled, so they were presented to the participant for evaluation in a totally random order.

5.3.5 Rule Evaluation

The sample rules for each participant were imported to the database of the online evaluation system. The participant was notified by email that the evaluation was ready to begin, and he/she could log into the system and click the *Evaluate rules* link from the menu to open the rules for evaluation.

Before the rules were presented for evaluation, an instruction page was displayed for the user. This page had two types of information: instructions on rule evaluation and instructions on system usage. Rule evaluation instructions demonstrated to the participant what an association rule was, from what aspects the rules should be evaluated, and what each aspect meant. The definition of association rule was explained with an example: *Bread, Butter* --> *Milk*, which described the fact that many of the customers who purchased *Bread* and *Butter* also purchased *Milk*. Similarly, the association rules presented for evaluation described certain associations between concepts in the participant's research area.

Participants were asked to evaluate the rules from three aspects: *plausibility*, *novelty* and *usefulness*. *Plausibility* means whether the established relationship sounds reasonable to the participant, *novelty* is to what extent the rule reveals something that is currently unknown to the participant in his/her research areas, and *usefulness* is to what degree the association rule helps the participant acquire new knowledge or understand existing concepts better in his/her research areas.

The system usage instruction showed the participant how to evaluate the rules and how to save the results. The participant needed to rate each evaluation aspect in a 7-point Likert scale, with 1 being the least and 7 being the most. The scale was implemented as a dropdown box with 8 items: empty and 1 to 7. Empty meant the rule had not been evaluated yet. The participant chose the most appropriate score from the dropdown list for each evaluation aspect. The interface is shown in Figure 5.7.

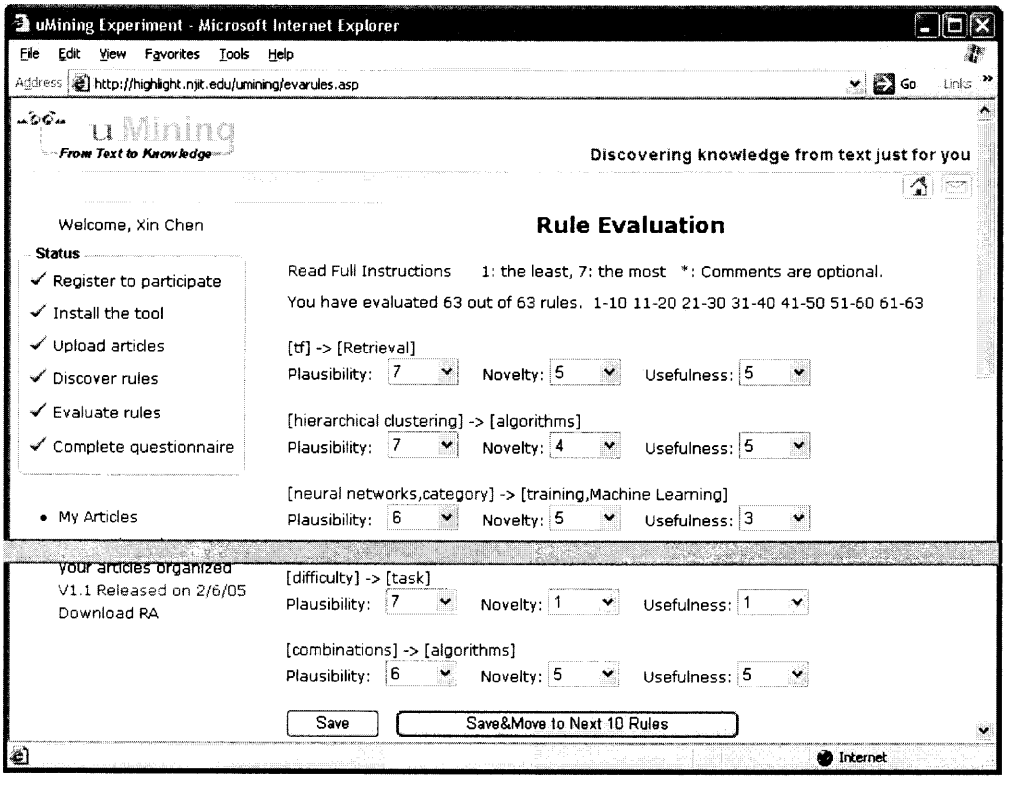


Figure 5.7 Rule evaluation interface.

Each page displayed ten rules for evaluation, thus the 63 sample rules were divided into seven pages. The participant was able to navigate between pages by clicking the number range links. After evaluating part or all of the rules in one page, the participant could click the *Save* button to save the current results. If the participant could not finish evaluating all the rules during one login session, he/she could continue the

evaluation from the last point when the results were saved. The participant could also click the *Save&Move to Next 10 Rules* button to save the evaluation results and open the next page for evaluation. The system displayed a congratulation message when all the 63 rules had been evaluated, and activated the *Complete questionnaire* link in the menu to allow the participant to answer the post-evaluation questionnaire.

The post-evaluation questionnaire (see Appendix D) had ten 7-point Likert scale questions and four open questions. The questionnaire was designed to obtain the subjective ratings of the participant about his/her confidence in the evaluation, the rule usefulness, and ease-of-use of the evaluation system. The open questions also allowed the participant to express more about his/her opinions about the rules and the evaluation.

The investigator was notified by the system once a participant finished evaluating the rules and completing the post-evaluation questionnaire. The evaluation results were downloaded from the online database for further analysis (see chapter 6).

5.4 Summary

This chapter presents the methodology of the user evaluation on the proposed text mining system. It starts with the research questions that need to be answered in the evaluation, followed by the pilot study and the preliminary results. The formal user study, including participant recruitment, system setup in the user study and the online evaluation system, is described.

CHAPTER 6

DATA ANALYSIS AND RESULTS

From the formal user study, two types of data were collected: the results calculated by the system and the subjective ratings submitted by the participants. This chapter presents the data analysis methods and the major findings from the analyses, especially the answers to the research questions brought up in Chapter 5.

6.1 Demographic Information of the Participants

In the background questionnaire, the demographic information of participants was collected. Such information gives an overview of all participants in this study, and it may also allow a better understanding of the results. There were 35 people registered in the online user study system, and 25 of them completed the entire study. Only those who completed the study are considered actual participants in the study. Table 6.1 shows the demographic information of all participants.

Table 6.1 Demographic Information of Participants

Characteristics	Type	Frequency	Percentage
Gender	Male	11	44%
	Female	14	56%
Stage of research	Looking for a topic	3	12%
	Completing SOTA	3	12%
	Completing proposal	9	36%
	Completing dissertation	8	32%
	Finished PhD study	2	8%
Native language	English	3	12%
	Non-English	22	88%
Computer Usage (hours per day)	<4	2	8%
	4 ~ 6	4	16%
	6 ~ 8	10	40%
	>8	9	36%
Text-mining system experience	Yes	7	28%
	No	18	72%

Table 6.1 shows that the gender of all participants in the study is roughly balanced. The majority of the participants are in the stages of completing their PhD dissertations. Only three participants are still looking for a topic, whereas the others are either narrowing their research topic (completing SOTA), or working on a specific topic (completing proposal and dissertation or finished PhD study).

6.2 System Settings and Results

The knowledge discovery process includes background knowledge development, target documents retrieval, feature extraction from target documents, association rules mining, and novelty calculation. This section discusses the input parameters to and output results from each component.

6.2.1 Parameters and System Settings

The parameters and system settings are summarized in Table 6.2.

Table 6.2 Summary of the Input Parameters and Output

P#	BGD	DF	N	H	TD	TF.IDF	Supp.	Conf.
1	34	7	0.8	14	393	20%	8	60%
2	51	10	0.8	9	318	20%	6	60%
3	63	13	0.8	8	270	20%	6	60%
4	69	14	0.8	9	313	20%	6	60%
5	41	8	0.8	8	327	20%	7	60%
6	22	4	0.8	13	532	20%	10	60%
7	98	20	0.8	10	515	20%	10	60%
8	26	5	0.8	12	612	20%	12	60%
9	39	8	0.8	14	474	20%	9	60%
10	50	10	0.8	11	274	20%	6	60%
11	41	8	0.8	10	374	20%	8	60%
12	48	10	0.8	12	413	20%	8	60%
13	41	8	0.8	9	401	20%	8	60%
14	61	12	0.8	12	343	20%	7	60%
15	14	3	0.8	9	500	20%	10	60%
16	39	8	0.8	8	496	20%	10	60%
17	30	6	0.8	13	446	20%	9	60%
18	55	11	0.8	10	584	20%	12	60%
19	29	6	0.8	13	521	20%	10	60%
20	32	6	0.8	14	513	20%	10	60%
21	38	8	0.8	11	534	20%	10	60%
22	12	2	0.8	7	630	20%	12	60%
23	53	11	0.8	8	564	20%	12	60%
24	38	8	0.8	7	520	20%	10	60%
25	24	5	0.8	11	607	20%	12	60%

The columns in Table 6.2 are described as follows.

P#: participant identification number.

BGD: the number of background documents provided by each participant.

DF: the document frequency threshold used for concept hierarchy development. Only the keywords that appear in more than DF background documents will be included in the concept hierarchy. Because participants have different numbers of background

documents, the DF threshold is set to be 20% of the total number of background documents, and the result is rounded to the nearest integer. These variables and their meanings are explained below.

N: the co-occurrence probability threshold. The POCA technique states that term *X* is the parent of term *Y* if and only if $P(X|Y) > P(Y|X)$ *and* $P(X|Y) >= N$, where $0 < N <= 1$ (Wu, 2001). The N threshold allows certain exceptions of *X*'s subsumption of *Y*. In previous studies (Sanderson and Croft, 1999; Wu, 2001), this threshold was set to 0.8. The same value was chosen for it in this study.

H: the depth (or height) of the generated concept hierarchy. It is the number of nodes in the longest path from the root to any leaf in the concept hierarchy. H is used in the normalization of the hierarchy distance calculation. The result shows that H ranges from 7 to 14, which indicates that the concept hierarchy is fairly rich to represent the semantic relationships among keywords in the background documents.

TD: the number of target documents in plain text format. This number is usually less than the total number of the PDF files downloaded from Google Scholar, because some of the PDF files are not convertible to text files for various reasons (see section 5.3.3.2 in Chapter 5 for details). TD ranges from 270 to 630. Because these articles were retrieved from Google Scholar with queries formulated from a participant's research interest(s), they can be considered relevant to the participant's research area(s). Compared with the number of background documents submitted by each participant, the size of target documents are significantly larger. Therefore, the target documents can be considered large enough to cover the major topics within the participant's research areas,

and from them it is possible to discover unknown and useful knowledge for the participant.

TF.IDF: the threshold for removing low ranked features (by their TF.IDF weights) from target documents. After noun phrases were extracted from target documents, they are weighed by their TF.IDF values. Within each target document, the noun phrases were ranked by their TF.IDF values in descending order, and the low 20% noun phrases were removed. Noun phrases with low TF.IDF scores are usually common terms in the participant's research areas, and they do not distinguish the content of one document from the others'.

Supp. is the support constraint, and Conf. is the confidence constraint applied in association rules mining. Support was set to 2% of the number of target documents, and confidence was set to 60%. A support constraint of 2% is low for traditional association rules mining from transaction databases, but for text mining it allows the system to capture some interesting relationships between concepts. Results from the pilot study show that association rules with high support values tend to contain general concepts and the relationships revealed by these rules are usually superficial. On the other hand, the low support association rules usually contain domain specific concepts and reflect particular relationships among these concepts, so they could be interesting to the user as well. Another reason for choosing a low support constraint was that only noun phrases were extracted as document features for association rules mining. Many domain specific noun phrases had low document frequency, so a low support threshold increased the chances to identify association rules among these noun phrases.

6.2.2 Discovered Association Rules

For each participant, the system discovered tens of thousands of association rules. The novelty of each rule was calculated and normalized to 1 to 7, with 1 being the least and 7 being the most. The number of association rules at each novelty level was counted. The total number of rules and the distribution of rules by novelty level are summarized in Table 6.3.

Table 6.3 Number of Discovered Association Rules

P#	# of Rules	L7 # %	L6 # %	L5 # %	L4 # %	L3 # %	L2 # %	L1 # %
1	14,049	218 1.6%	259 1.8%	952 6.8%	4,694 33.4%	4762 33.9%	1,589 11.3%	1,575 11.2%
2	37,060	591 1.6%	630 1.7%	3,723 10.0%	10,214 27.6%	9,018 24.3%	7,833 21.1%	5,051 13.6%
3	31,820	367 1.2%	632 2.0%	2,176 6.8%	5,320 16.7%	12,027 37.8%	7,075 22.2%	4,223 13.3%
4	16,150	167 1.0%	353 2.2%	1,162 7.2%	3,460 21.4%	4,194 26.0%	4,113 25.5%	2,701 16.7%
5	42,990	286 0.7%	1,212 2.8%	5,361 12.5%	7,923 18.4%	13,557 31.5%	9,638 22.4%	5,013 11.7%
6	25,574	150 0.6%	816 3.2%	2,422 9.5%	8,096 31.7%	5,313 20.8%	4,411 17.2%	4,366 17.1%
7	49,836	515 1.0%	1,136 2.3%	5,510 11.1%	10,579 21.2%	14,074 28.2%	10,022 20.1%	8,002 16.1%
8	25,388	444 1.7%	836 3.3%	2,624 10.3%	4,104 16.2%	9,704 38.2%	3,258 12.8%	4,418 17.4%
9	15,613	212 1.4%	221 1.4%	1,510 9.7%	3,021 19.3%	4,464 28.6%	3,506 22.5%	2,679 17.2%
10	15,156	124 0.8%	260 1.7%	1,170 7.7%	2,863 18.9%	4,461 29.4%	3,262 21.5%	3,016 19.9%
11	38,675	396 1.0%	711 1.8%	3,660 9.5%	7,176 18.6%	14,459 37.4%	6,150 15.9%	6,123 15.8%
12	27,675	314 1.1%	814 2.9%	3,603 13.0%	7,701 27.8%	7,935 28.7%	4,547 16.4%	2,761 10.0%
13	20,170	278 1.4%	564 2.8%	2,910 14.4%	6,620 32.8%	3,903 19.3%	3,072 15.2%	2,825 14.0%
14	14,571	144 1.0%	315 2.2%	1,686 11.6%	2,155 14.8%	4,241 29.1%	3,506 24.1%	2,524 17.3%
15	11,692	157 1.3%	263 2.2%	1,058 9.0%	3,103 26.5%	2,521 21.6%	2,176 18.6%	2,414 20.6%

Table 6.3 Number of Discovered Association Rules (Continued)

P#	# of Rules	L7 # %	L6 # %	L5 # %	L4 # %	L3 # %	L2 # %	L1 # %
16	28,595	210 0.7%	640 2.2%	3,879 13.6%	5,089 17.8%	6,198 21.7%	7,300 25.5%	5,281 18.5%
17	33,285	228 0.7%	1,004 3.0%	2,355 7.1%	11,159 33.5%	7,651 23.0%	6,248 18.8%	4,640 13.9%
18	40,040	475 1.2%	1,031 2.6%	2,943 7.4%	11,934 29.8%	9,865 24.6%	8,411 21.0%	5,381 13.4%
19	13,627	131 1.0%	429 3.1%	765 5.6%	4,013 29.5%	4,639 34.0%	2,243 16.5%	1,405 10.3%
20	33,162	431 1.3%	870 2.6%	3,337 10.1%	6,465 19.5%	6,331 19.1%	9,932 29.9%	5,796 17.5%
21	27,559	468 1.7%	453 1.6%	1,562 5.7%	8,140 29.5%	7,799 28.3%	4,534 16.5%	4,603 16.7%
22	10,108	136 1.3%	363 3.6%	439 4.3%	3,236 32.0%	2,754 27.2%	1,750 17.3%	1,432 14.2%
23	41,285	424 1.0%	1,240 3.0%	3,847 9.3%	10,562 25.6%	1,3929 33.7%	4,275 10.4%	7,008 17.0%
24	39,502	332 0.8%	678 1.7%	4,582 11.6%	8,344 21.1%	10,793 27.3%	8,091 20.5%	6,682 16.9%
25	21,749	254 1.2%	598 2.7%	1,962 9.0%	6,421 29.5%	5,701 26.2%	4,675 21.5%	2,138 9.8%
	Mean	298.08 1.1%	653.12 2.4%	2,607.92 9.3%	6,495.68 24.5%	7,611.72 28.0%	5,264.68 19.4%	4,082.28 15.2%

Note: L1~7: novelty level one to level seven.

The average number of association rules at each novelty level is calculated in the last row of Table 6.3. The following figure shows the distribution of the average number of rules by novelty level.

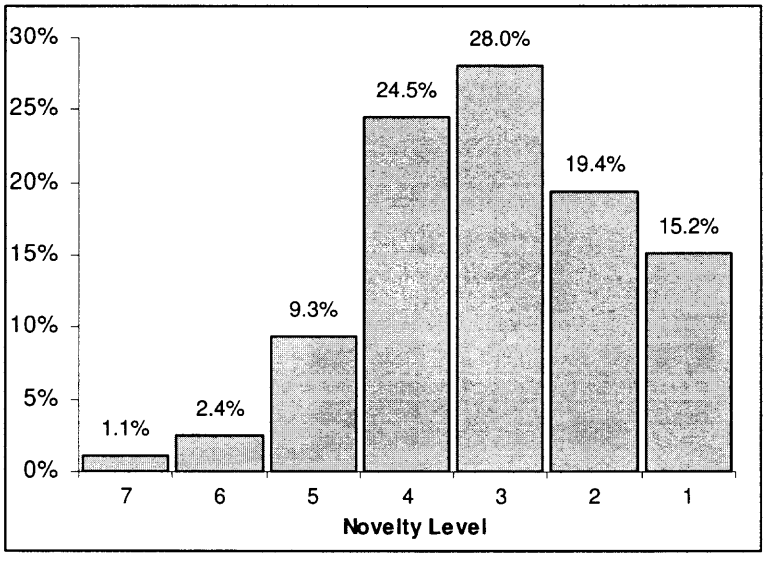


Figure 6.1 Rule distribution by novelty level.

Figure 6.1 shows that more than 60% of the rules fall into the range from level one to level three, and only a few rules (less than 20%) are at novelty level five to seven. If rules at level five to seven are considered novel and those at level one to three are considered non-novel (level four is neutral), majority of the rules are rated not novel by the system. If the system prediction is accurate, then the system can greatly help users find real novel rules. The following sections will investigate the performance of the user-oriented novelty measure in terms of identifying interesting (novel and useful) rules.

6.3 Performance of the Novelty Measure

The goal of a Data Mining system is to find previously unknown and potentially useful patterns, thus the performance of the user-oriented novelty measure is evaluated from two perspectives: predicting novel (previously unknown) rules and identifying useful rules. The analyses provide answers to research questions no. 1 and no. 2 in Chapter 5. Before

the performance evaluation is discussed, the format of the raw data is described first for easy understanding of the data used in the performance evaluation.

6.3.1 Format of the Raw Data

Raw data consists of two parts: results calculated by the system and data submitted by the participants. They are associated with the samples rules for every participant. Each participant was asked to rate the novelty and usefulness of the samples rules, so each rule had a subjective novelty score (*SN*) and a subjective usefulness score (*SU*). In addition to *SN* and *SU*, for each rule the program also calculated eight objective measures: Support (*S*), Confidence (*C*), Odds ratio (α), Jaccard (ζ), Piatetsky-Shapiro's (*PS*), Gini Index (*G*), Klosgen (*K*) and Kappa (κ), as well as the WordNet novelty measure (*WN*) and the user-oriented novelty measure (*UN*), as shown in Figure 6.2.

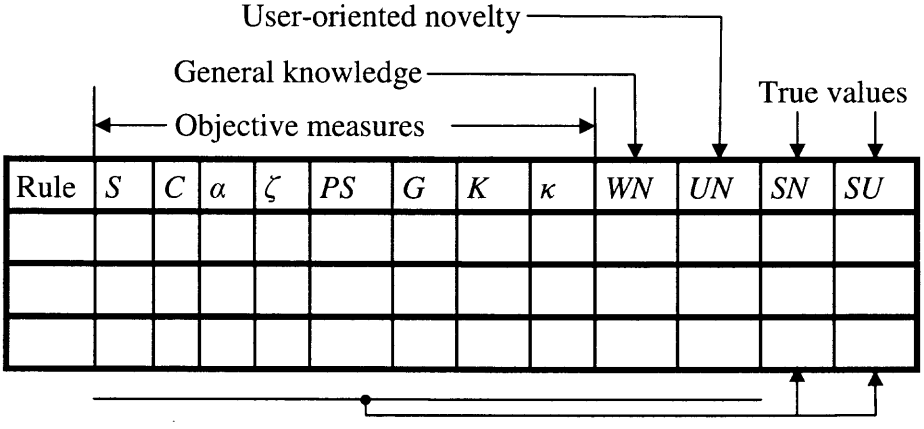


Figure 6.2 Format of the raw data.

Figure 6.3 shows the format of the raw data for one participant. Each row corresponds to one sample rule. It consists of all the scores of the measures mentioned above. *SN* and *SU* can be considered the true values of the novelty and the usefulness of association rules for a participant respectively. Therefore, the scores of different

measures (in different columns) can be compared to the true values (*SN* for novelty and *SU* for usefulness) to investigate which measures are closer to the user ratings, i.e. having better performance.

6.3.2 Novelty Prediction Accuracy

Novelty prediction accuracy analysis seeks answers to research question Q1.1 (is the novelty prediction accurate?) and Q2.1 (does the user-oriented novelty measure perform better than other novelty measures in predicting the novelty of association rules?). For a given measure, its novelty prediction accuracy was evaluated by comparing the scores of this measure with the actual user subjective novelty ratings. The comparison was performed on the correlations derived from the raw data. Correlation between *UN* and *SN* was computed to verify if there is a strong association between them: the larger the correlation, the stronger the association and the more accurate the prediction. *UN* was also compared with *WN* for novelty prediction.

The Pearson product-moment correlations between the subjective novelty ratings (*SN*) and the four measures – Support (*S*), Confidence (*C*), the WordNet novelty (*WN*) and the user-oriented novelty (*UN*) measure – are shown in Table 6.4.

Basu et al (2001) used all four categories of words in WordNet to calculate the rule novelty, but in this study the association rules were discovered among noun phrases extracted from documents. Therefore, the verb and the adverb categories were eliminated from the WordNet hierarchy when *WN* was calculated. The results show that among all interestingness measures, *UN* has the highest correlation with *SN*. Since the subjective ratings can be viewed as the true values of the rule novelty to the corresponding user, *UN*

then has the highest prediction accuracy. The following figure provides another view of the Pearson product-moment correlation coefficients (r) of the four measures with *SN*.

Table 6.4 Correlations between Four Measures and *SN*

P#	Correlations (r)			
	S-SN	*C-SN*	*WN-SN*	*UN-SN*
1	-0.09	0.18	0.38	0.68
2	-0.13	0.17	0.11	0.53
3	-0.01	-0.05	-0.07	0.31
4	-0.39	0.10	0.43	0.56
5	0.21	-0.12	-0.04	-0.05
6	-0.37	0.05	0.25	0.45
7	-0.08	0.10	0.02	0.07
8	-0.40	-0.04	0.04	0.47
9	-0.14	0.20	0.16	0.19
10	-0.11	0.04	0.05	0.23
11	0.02	-0.03	0.33	0.41
12	-0.03	-0.02	-0.10	0.21
13	-0.26	0.27	0.11	0.28
14	-0.09	-0.21	0.20	0.28
15	0.01	0.05	0.20	0.28
16	-0.13	-0.18	0.00	0.07
17	-0.14	-0.13	0.36	0.37
18	-0.10	0.01	-0.03	0.40
19	-0.22	0.08	0.11	0.40
20	-0.25	-0.04	0.31	0.68
21	-0.38	-0.28	0.34	0.59
22	-0.39	0.21	0.30	0.71
23	-0.25	0.27	0.33	0.55
24	-0.19	0.16	0.11	0.41
25	-0.12	0.05	0.20	0.38
Mean	-0.161	0.034	0.164	0.378
Std.	0.151	0.147	0.156	0.197

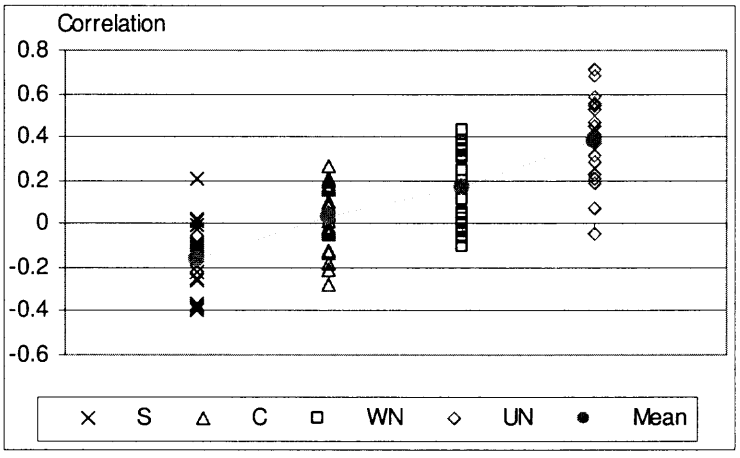


Figure 6.3 Novelty prediction accuracy (correlations with true value *SN*).

Table 6.4 and Figure 6.3 show that *S* in general has a negative correlation with *SN*. It can be explained that a high Support score means that the associated concepts occur frequently in the documents, and the corresponding relationships between concepts tend to be superficial, so the rule itself is less likely to be novel to the user. For example, for one participant in the Text Mining area, the system discovered the rule, *document -> text*, which has a support value of 0.23 (or 23%). This is a well-known relationship between two general concepts in this area, so it is not novel at all to the user (the novelty of this rule was rated 1 by the participant).

Confidence measures the strength of a rule; it is the conditional probability of the consequent given the presence of the antecedent. It has a higher correlation with *SN* than *S* does, but the correlation is still close to zero. The WordNet novelty measure has a higher correlation with *SN* than both *S* and *C* do. It can be explained that the WordNet database is developed manually by human experts, and it does capture the semantic relationships among concepts, from a universal point of view. Finally, the user-oriented novelty measure has the largest correlation among all four measures.

One remark that needs to be made is the reason to include *S* and *C* in the above comparison. Since *S* and *C* are not designed for identifying novel rules, it may not be fair to compare them with the *WN* and *UN* measures for the purpose of identifying novel rules. However, they can be used as a baseline, and any novelty measures should at least perform better than these two measures. The result is as expected, and it can be concluded that the *WN* and *UN* measures predict the novelty of an association rule not merely by chance, thus it makes sense to compare *WN* and *UN* for their performance in identifying novel association rules.

Significance tests were run to investigate whether it was the types of measures that caused the difference in the performance (correlations between the measures and the user ratings). Since correlation data are not normally distributed, they do not satisfy the assumptions of parametric tests (e.g., T-tests and F-tests). The non-parametric Kruskal-Wallis test and Wilcoxon test were chosen for significance tests.

Wilcoxon rank-sum test is the non-parametric equivalent of the two sample t-test, and it can be used to test the null hypothesis that two samples drawn from similarly-shaped distribution are drawn from the same distribution, against the alternative hypothesis that the populations have different locations. Kruskal-Wallis test is the generalized form of the Wilcoxon test, and it is usually used when the categorical variable has more then two values. The Kruskal-Wallis test was first run to test if there was any difference in the performance of difference measures. The results are show in Table 6.5.

Table 6.5 Result of the Kruskal-Wallis Test

Measure	N	Sum of Scores	Expected Under H0	Std Dev Under H0	Mean Scores
S	25	1904.00	1262.50	125.623	76.16
C	25	1416.00	1262.50	125.623	56.64
WN	25	1052.50	1262.50	125.623	42.10
UN	25	677.50	1262.50	125.623	27.10
Kruskal-Wallis Test Chi-Square 39.0375 DF 3 Pr > Chi-Square <0.0001					

Note: average scores were used for ties.

The test is significant at $p<0.0001$ level, thus the null hypothesis that there is no difference in the performance of different measures can be rejected. Five pairwise Wilcoxon tests (*WN* vs. *S*, *WN* vs. *C*, *UN* vs. *S*, *UN* vs. *C*, and *UN* vs. *WN*) were conducted to further investigate which measures had better performance in novelty prediction. The results are shown in Table 6.6.

Table 6.6 Wilcoxon Tests on Novelty Prediction Accuracy

Comparison	N	DF	Statistic	Normal Approximation	
				Z	Pr > Z
WN vs. S	25	1	864.00	4.385	<0.001
WN vs. C	25	1	744.00	2.057	0.020
UN vs. S	25	1	901.00	5.103	<0.001
UN vs .C	25	1	836.00	3.842	<0.001
UN vs. WN	25	1	760.50	2.377	0.0087

The results show that, for identifying novel association rules, both *WN* and *UN* perform significantly better than the baseline measures *S* and *C*, and *UN* performs significantly better than *WN*. In other words, *UN* performs the best in predicting the novelty of association rules.

6.3.3 Novelty vs. Usefulness

Analyses in this section will try to find the answers to research questions Q1.2 (are novel association rules also useful to users?) and Q2.2 (does the user-oriented novelty measure perform better than other interestingness measures in terms of identifying useful association rules?).

Interestingness measures evaluate the interestingness of association rules from different aspects, and it is possible that an interestingness measure can identify totally "interesting" but not useful rules. In other words, interestingness does not guarantee usefulness. Since one of the important goals of a data mining system is to find useful rules, a good interestingness measure is also expected to identify useful rules for the user. This section will explore the relationship between novelty and usefulness of association rules, and investigate whether novelty measures are good usefulness indicators. Since usefulness cannot be directly measured by any of the measures, the ability of finding useful rules of an interestingness measure is called the usefulness indication power.

Similarly, evaluating the usefulness indication power of *UN* includes measuring the correlation between *UN* and *SU*, and comparing *UN* against other interestingness measures (e.g., *WN* and objective measures) for indicating useful rules. In addition to *S* and *C*, various objective measures have been used for evaluating the interestingness of association rules. In (Tan et al. 2004), a total of 21 objective measures were studied, and the representative measures were classified into 7 groups according to their property similarities, as shown in Table 6.7.

Table 6.7 Groups of Objective Measures

Group	Objective interestingness measures
1	Odds ratio (α), Yule's Q, Yule's Y
2	Cosine (IS), Jaccard (ζ)
3	Support (S), Laplace (L)
4	Φ-coefficient, Collective strength (S), Piatetsky-Shapiro's (PS)
5	Gini Index (G), Goodman-Kruskal's (λ)
6	Interest (I), Added Value (AV), Klosgen (K)
7	Mutual Information (M), Certainty factor (F), Kappa (κ)

Since the measures in each group share similar properties, one measure from each group was chosen for comparison. The chosen objective measures were Support (S), Odds ratio (α), Jaccard (ζ), Piatetsky-Shapiro's (PS), Gini Index (G), Klosgen (K) and Kappa (κ).

6.3.3.1 Correlations. Table 6.8 shows the correlations between all measures and SU.

Table 6.8 Correlations between All Measures and SU for Usefulness Indication

	Correlations (r)									
P#	*S-SU*	*α-SU*	*ζ-SU*	*PS-SU*	*G-SU*	*K-SU*	*κ-SU*	*WN-SU*	*UN-SU*	*SN-SU*
1	-0.02	0.06	0.05	0.10	0.09	0.07	0.12	0.40	0.65	0.85
2	-0.03	-0.14	-0.16	0.01	-0.04	-0.12	-0.16	0.05	0.12	0.50
3	-0.10	-0.04	-0.17	-0.06	-0.05	0.00	-0.10	0.02	0.06	0.31
4	-0.20	0.08	0.04	-0.19	-0.13	-0.19	-0.05	0.12	0.21	0.56
5	-0.13	0.17	0.03	-0.08	-0.04	-0.09	0.09	-0.09	-0.19	0.34
6	-0.21	0.12	0.16	0.09	0.14	-0.05	0.19	0.16	0.26	0.46
7	-0.05	-0.11	0.02	-0.14	-0.18	0.12	0.06	0.00	-0.03	0.78
8	0.13	-0.11	-0.19	0.21	0.20	-0.14	0.12	0.12	0.06	-0.23
9	-0.19	0.24	0.13	-0.08	0.10	0.16	0.16	0.24	0.31	0.82
10	-0.03	-0.07	-0.06	0.08	0.10	0.10	-0.01	0.13	0.01	0.54
11	0.03	0.12	-0.09	-0.14	-0.01	0.02	0.14	0.37	0.47	0.67
12	0.17	0.02	-0.13	0.19	0.14	-0.11	-0.16	-0.22	-0.23	0.24
13	-0.01	0.19	0.08	0.16	0.23	-0.03	0.10	-0.01	0.17	0.53
14	-0.04	-0.01	0.09	-0.21	-0.11	0.20	0.05	-0.14	-0.11	-0.13
15	-0.06	-0.12	-0.04	-0.08	0.13	0.02	-0.28	0.36	0.11	0.54
16	0.00	-0.06	-0.06	0.13	0.15	-0.04	-0.05	0.10	-0.02	0.33
17	0.28	-0.15	-0.08	0.23	0.21	-0.15	-0.19	-0.27	-0.45	-0.03
18	0.09	-0.10	0.14	0.28	0.40	0.40	0.12	0.09	-0.31	-0.29
19	-0.30	0.04	-0.03	0.14	0.14	0.02	0.04	0.10	0.33	0.89
20	-0.19	-0.11	-0.22	-0.22	-0.19	-0.19	-0.14	0.28	0.71	0.93

Table 6.8 Correlations between All Measures and *SU* for Usefulness Indication (Continued)

	Correlations (r)									
P#	*S-SU*	*α-SU*	*ζ-SU*	*PS-SU*	*G-SU*	*K-SU*	*κ-SU*	*WN-SU*	*UN-SU*	*SN-SU*
21	-0.47	0.12	-0.13	-0.41	-0.32	-0.27	0.08	0.37	0.60	0.89
22	-0.43	0.16	0.13	-0.50	-0.33	-0.40	0.24	0.33	0.42	0.69
23	-0.31	0.25	-0.13	-0.12	-0.13	-0.05	0.13	0.33	0.43	0.70
24	-0.27	0.12	0.02	-0.25	-0.04	0.28	0.14	0.01	0.39	0.74
25	-0.04	0.03	-0.10	0.13	-0.05	0.20	0.22	0.25	0.32	0.44
Mean	-0.095	0.028	-0.028	-0.029	0.016	-0.010	0.034	0.124	0.172	0.483
Std.	0.179	0.125	0.112	0.202	0.177	0.178	0.140	0.188	0.301	0.351

The last column of Table 6.8 is the correlation between *SN* and *SN*, which reflects how participants treat the relationship between novelty and usefulness of association rules. The results show that majority of the participants have a high correlation between the two measures, which suggests that in most cases novel rules are also thought to be useful, and novelty measures (e.g. *WN* and *UN*) can be good candidate measures for identifying useful rules. However, there are four participants who have negative correlations between *SN* and *SU* (highlighted in the last column of Table 6.8). The negative correlations suggest that the four participants do not think novel (unknown) rules are useful. For such participants, novelty measures will no longer be good usefulness indicators, no matter how accurate they are for novelty prediction. Since more than 80% of the participants have positive correlations between *SN* and *SU*, for them novelty measures are good usefulness indicators; therefore it is still worthwhile investigating the usefulness indication power of UN and comparing it with other interestingness measures.

The mean values of the correlations at the bottom of Table 6.8 show that, other than $r_{SN\text{-}SU}$, *UN* generally has a higher correlation with *SU* than other measures, which suggests that *UN* performs better than other interestingness measures in identifying useful

association rules. *WN* has a higher correlation than all objective measures as well, which suggest that taking into consideration of general knowledge helps in finding useful rules. However, the correlation of *WN* is smaller than that of *UN*. Figure 6.4 visualizes the correlations and their mean values of all measures.

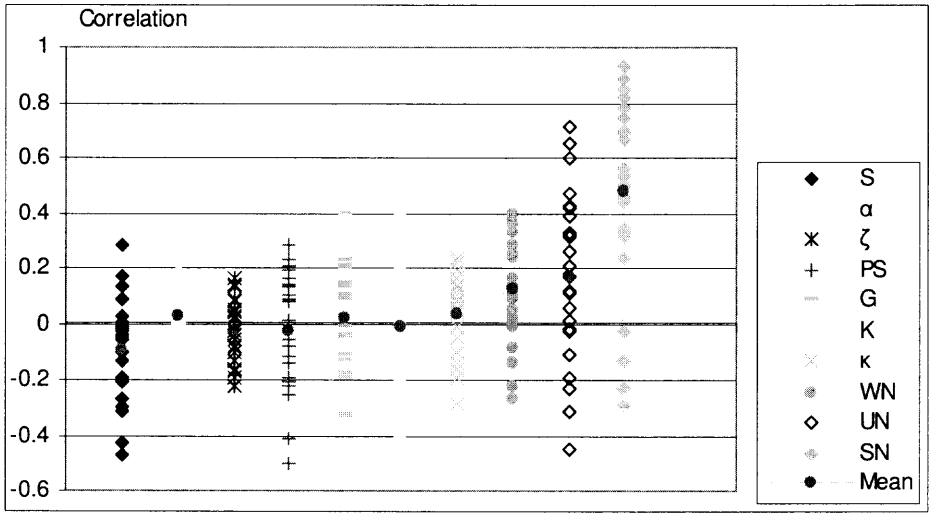


Figure 6.4 Usefulness indication power.

The objective measures have low correlations, because they do not take advantage of any general/domain knowledge or the participants' background knowledge. The mean values in Table 6.8 and Figure 6.4 give a general idea of which measures have higher correlations with the true usefulness ratings. A series of non-parametric significance tests were conducted to investigate whether there were significant differences in the performance of interestingness measures for identifying useful association rules, and if yes, which measures had better performance than others. The pairwise comparisons were made among *UN*, *WN* and each of the seven objective measures. The results are shown in Table 6.9.

Table 6.9 Result of the Kruskal-Wallis Test on Usefulness Indication Power

Measure	N	Sum of Scores	Expected Under H0	Std Dev Under H0	Mean Scores
S	25	1959.50	3137.50	342.94	78.38
α	25	2984.00	3137.50	342.94	119.36
ζ	25	2421.50	3137.50	342.94	96.86
PS	25	2575.50	3137.50	342.94	103.02
G	25	2925.50	3137.50	342.94	117.02
K	25	2624.50	3137.50	342.94	104.98
κ	25	3094.00	3137.50	342.94	123.76
WN	25	3783.50	3137.50	342.94	151.34
UN	25	3884.50	3137.50	342.94	155.38
Kruskal-Wallis Test Chi-Square 57.1302 DF 9 Pr > Chi-Square <0.0001					

Note: average scores were used for ties.

The test is significant at $p<0.0001$ level, thus the null hypothesis that there is no difference in the performance of different measures in identifying useful rules can be rejected.

Table 6.10 Wilcoxon Tests on Usefulness Indication Power

Comparison	N	DF	Statistic	Normal Approximation	
				Z	Pr > Z
WN vs. S	25	1	444.00	3.746	<0.0001
WN vs. α	25	1	536.00	1.962	0.028
WN vs. ζ	25	1	485.50	2.941	0.002
WN vs. PS	25	1	517.50	2.320	0.010
WN vs. G	25	1	541.00	1.864	0.031
WN vs. K	25	1	511.00	2.447	0.009
WN vs. κ	25	1	551.50	1.660	0.048
UN vs. S	25	1	462.50	3.387	0.001
UN vs. α	25	1	532.00	2.039	0.023
UN vs. ζ	25	1	495.00	2.756	0.004
UN vs. PS	25	1	510.50	2.455	0.007
UN vs. G	25	1	528.50	2.106	0.018
UN vs. K	25	1	511.50	2.436	0.009
UN vs. κ	25	1	543.00	1.825	0.037
UN vs. WN	25	1	604.50	0.631	0.266

More pairwise Wilcoxon tests were conducted to further investigate which measures (objective measures vs. *WN* vs. *UN*) had better performance in usefulness indication. The results are shown in Table 6.10. In general, the novelty measures (*WN* and *UN*) perform significantly better than the objective measures in terms of correlating with user ratings. *UN* has a higher correlation with *SU* than *WN* does, but the difference is not significant. The results suggest that novelty of an association rule to some extent reflects its usefulness, so novelty measures (*WN* and *UN*) are more correlated with the subjective rule usefulness ratings than objective measures. However, novelty alone does not guarantee usefulness, thus even though *UN* performs significantly better than *WN* in predicting the novelty of association rules, the difference between their performance in identifying useful rules is not significant.

6.4 Analysis of Affecting Factors

One of the key parameter in background knowledge development is the number of background documents provided by the user. Analysis in this section will investigate the effect of the number of background documents on the performance *SN*. The effects of user factors, such as gender and current stage of research, on the system performance are also studied. The analyses will provide the answers to research question Q3 (what are the factors affecting the performance of the user-oriented novelty measure?).

6.4.1 Effects of the Number of Background Documents

In order to obtain the main concepts and to capture their relationships in the user's background knowledge, the system needs a certain number of background documents (*#BGD*). This number will affect not only the quality of the background knowledge, but

also the user acceptance of the system. Too few background documents may not be able to cover the key concepts in the user's background knowledge, so the keyword space developed from the background documents may not be representative for the user's knowledge. However, if the system requires too many background documents from the user in order to develop a representative background keyword space, it may prevent the user from using the system. Therefore, it is necessary to understand the effect of *#BGD* on the measure performance for novelty prediction and usefulness indication. The question to be answered is how many background documents are needed to generate a reasonable system performance (Q3.1).

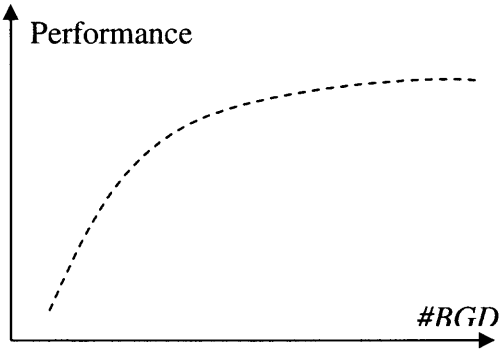


Figure 6.5 Expected effect of *#BGD* on the performance of *UN*

The dotted line in Figure 6.5 is the expected pattern for the effect of *#BGD* on the performance of *UN*. When *#BGD* is small, the performance (correlations) is expected to be low, because the background documents may not be representative for the user's knowledge. When *#BGD* increases, the performance increases too and eventually approaches to the ideal level the system can achieve. The actual relationship between *#BGD* and the measure performance (novelty prediction accuracy and usefulness indication power) are plotted in Figure 6.6. The X-axis is *#BGD* and the Y-axis is the performance (correlation).

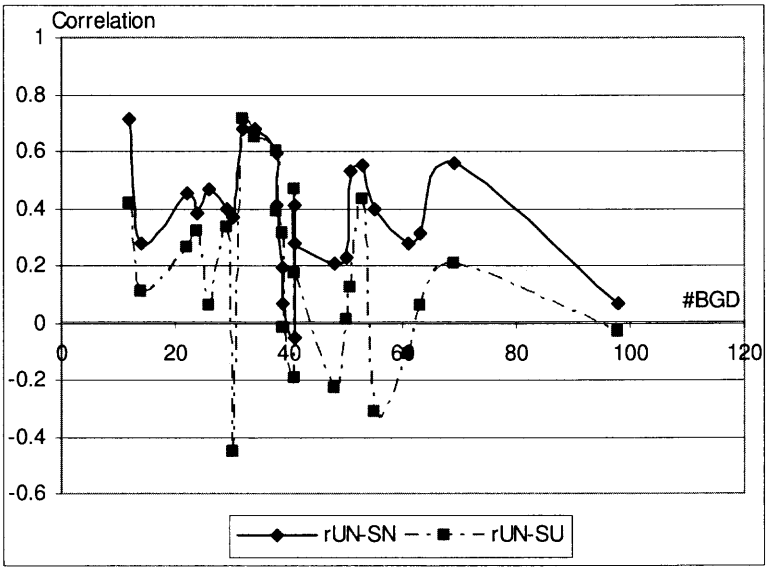


Figure 6.6 Effects of *#BGD* on the performance of *UN*.

However, the actual results do not show the expected pattern. There is no clear trend of the measure performance as *#BGD* increases. Figure 6.6 suggests that there is no clear relationship between the measure performance and *#BGD*, but it does not support that *#BGD* has no effect on the measure performance. To test the effect of *#BGD*, the participants were divided into three groups by converting *#BGD* into an ordinal variable with three values: small, medium and large. The cutoff points were chosen such that there were roughly equal participants in each group. A Kruskal-Wallis test was run to test if there was any difference in the measure performance across groups.

Table 6.11 Kruskal-Wallis Test on the Effect of *#BGD* on Novelty Prediction

Group	N	Sum of Scores	Expected Under H0	Std Dev Under H0	Mean Scores
Small	7	104.50	91.0	16.50	14.93
Medium	9	115.50	117.0	17.64	12.83
Large	9	105.00	117.0	17.64	11.67
Kruskal-Wallis Test Chi-Square 0.7831 DF 2 Pr > Chi-Square 0.6760					

Table 6.11 presents the result of the Kruskal-Wallis test. The result shows that there is no significant difference in the measure performance for novelty prediction across different groups. Similar result was found when the test was run on the measure performance for usefulness indication. Therefore, it can be concluded that *#BGD* does not have significant effect on the measure performance.

A closer look at the patterns in Figure 6.5, especially the one for $r_{UN\text{-}SU}$, shows that when *#BGD* is larger than 40, the correlations tend to jump around and vary more greatly than those with smaller *#BGD*. The possible reason could be that when a participant submits more background documents, they become less coherent. Lack of coherence in the background documents usually leads to low precision of the concept hierarchy (Wu, 2005), which may further affect the novelty calculation. For example, the participant having the largest *#BGD* submitted all articles used in his State-Of-The-Art literature review. This review is usually done before a PhD student chooses a particular topic for his/her dissertation, and the PhD student is expected to explore all related fields in the review and identify potential research topics. Because the articles cover many different topics, the concept hierarchy developed from the background documents may not be able to capture the correct relationships between keywords in a specific domain, and this in turn may affect the novelty prediction accuracy.

6.4.2 User Information

In the background questionnaire answered by participants when they signed up for this study, the demographic information of participants was collected. Table 6.1 shows that participants have different backgrounds. In previous analysis, *UN* has been compared with *WN* and other objective measures for their novelty prediction accuracy and

usefulness indication power. The difference between the performance of *UN* and that of other measures are significant. Since participants differ a lot in their background, a further analysis is needed to investigate whether such differences may influence the performance of the interestingness measures (research question Q3.2). This analysis will also provide a better understanding of the relationship between the measure performance and different types of users.

Multiple linear regression analyses were performed to investigate if the significant differences among the performance of interestingness measures remain after controlling the differences in other characteristics of the participants. The analyses were conducted for both novelty prediction and usefulness indication. Table 6.12 details the regression results for novelty prediction accuracy. Each cell in the table reports the regression coefficient, t value, and the probability that the coefficient is significantly different from zero. The explanatory variables of main interest are the types of novelty measures: *t_WN* (1 for the *WN* measure and 0 otherwise), and *t_UN* (1 for the *UN* measure and 0 otherwise). The Confidence measure is chosen as the baseline measure, so the two independent variables will show whether, on average, the novelty prediction accuracy of each measure is significantly different than the results from the Confidence measure, independent of differences in other user characteristics. User factors controlled in the regressions are Gender (1 for *male* and 0 for *female*), Native Language (1 for *English* and 0 for *other languages*), Research Stage (1 for *Looking for a topic*, 2 for *Completing the State-Of-The-Art literature review*, 3 for *Completing proposal*, 4 for *Completing dissertation*, and 5 for *Finished PhD study*), and Computer Usage (1 for *<4 hours per day*, 2 for *4~6 hours per day*, 3 for *6~8 hours per day*, and 4 for *>5 hours per day*).

Table 6.12 Regression Analysis: Effects of User Factors on Novelty Prediction

	Regression 1			Regression 2		
	Coefficients	t Value	p	Coefficients	t Value	p
T_WN	0.130	2.74	0.008	0.130	2.75	0.008
T_UN	0.345	7.25	<0.001	0.345	7.28	<0.001
Gender	X	X	X	0.040	0.96	0.340
Lang.	X	X	X	0.099	1.33	0.189
Stage	X	X	X	-0.001	-0.07	0.944
Comp.	X	X	X	-0.036	-1.50	0.137

Regression 1, used as a baseline, shows that not taking into account user factors, the novelty prediction accuracy of *WN* is on average 13% higher and the accuracy of *UN* is 34.5% higher when compared with the Confidence measure ($p<0.05$ for both coefficients). The coefficients of interests remain the same in regression 2 where controlled user factors are introduced into the regression model, and none of the controlled variables have significant effects on the difference in the novelty prediction accuracy of the two measures. Similar regressions were run for the usefulness indication power of *WN* and *UN*, except that Kappa (κ) is chosen as the baseline measure. The result is shown in Table 6.13.

Table 6.13 Regression Analysis: Effects of User Factors on Usefulness Indication

	Regression 1			Regression 2		
	Coefficients	t Value	p	Coefficients	t Value	p
WN	0.090	1.44	0.154	0.090	1.54	0.128
UN	0.137	2.21	0.031	0.137	2.36	0.021
Gender	X	X	X	0.107	2.07	0.042
Lang.	X	X	X	0.231	2.51	0.015
Stage	X	X	X	0.015	0.60	0.551
Comp.	X	X	X	-0.062	-2.09	0.040

The baseline regression (regression 1) shows that without considering user factors, the usefulness indication power of *WN* is on average 9% higher than that of κ, but the difference is not significant. *UN* is 13.7% higher than κ and the difference is significant at p=0.05 level. The two coefficients remain the same in regression 2 after controlling the user factors, even if some of the controlling variables have significant effects on the difference in the usefulness indication power.

The analyses suggest that the regression models are robust to all controlled variables (user factors). The significant differences between the performances of different interestingness measures are introduced by the measures, not user factors.

6.5 Post-questionnaire Analysis

After the rule novelty and usefulness evaluation, the participants were asked to complete a post-evaluation questionnaire, which consists of questions about the participants' opinion and experience about the results and the user study.

6.5.1 Closed Questions

These questions ask the participant's opinion and experience by providing them with 7-point Likert scales, from which the participants chose the most appropriate answers. See Appendix D for the post-evaluation questionnaire. Table 6.14 lists all the closed questions and the mean scores of the original answers from 25 participants. For each question, answer 1 means the least, and answer 7 means the most.

Table 6.14 Questions and Answers in the Post-questionnaire

Question	Frequency of Answers							Mean*	Std.
	1	2	3	4	5	6	7		
1. Expertise in research areas.		1	1	2	8	13		5.24	1.05
2. Background document relevancy			1	3	5	10	6	5.68	1.11
3. Association rule relevancy			1	4	5	12	3	5.48	1.05
4. Rules introduce new concepts	1	1	3	4	6	8	2	4.80	1.53
5. Rules introduce new relationships	1	1	2	7	8	6		4.52	1.29
6. Usefulness of discovered rules	1	1	3	3	4	8	5	5.08	1.68
7. Usefulness of the text mining system			1	3	5	8	8	5.76	1.16
8. Importance of finding new relationships			1	2	4	9	9	5.92	1.12
9. Clarity of study instructions		1	1	4	5	7	7	5.48	1.39
10. Ease of use of the evaluation system			1	1	5	7	11	6.04	1.10

*: all are significant at p=0.05 level when compared to the mid-point 4 of the scale using the one sample t-test.

Most of the participants are in the stage of completing proposal and dissertation, so they have been doing research in the related areas for some time. This is confirmed by the participant self-evaluation of their expertise in their research areas (mean score 5.24). Therefore, the participants should have the ability to judge the novelty and usefulness of association rules extracted from relevant documents. The background documents provided by participants are consider relevant to their research (mean score 5.68), so the assumption that the background documents are relevant to participants' research is satisfied.

The discovered association rules are considered relevant to a participant's research areas too (mean score 5.48), and it ensures that the novelty and usefulness judgment is not affected by the irrelevancy of the rules. When compared to the mid-point of the scale (4), all the means are significantly larger. Therefore, participants did perceive that the rules introduced new concepts and new relationships between concepts, and they were useful. Participants agree that it is important to identify new relationships

among concepts in their research areas, and a system with this ability will be very helpful. As to the user study, the instructions were clear, and the evaluation system was easy to use.

6.5.2 Open-ended Questions

The open-ended questions ask participants to explain more about their answers to the closed questions. For example, if a participant chose to answer that there were useful rules identified, he/she will be asked to explain more why and how the rules were useful to his/her research. Below are a few comments from the participants explaining the reason they thought the rules were useful.

- Many rules discovered during the evaluation correctly reflected the relationships between concepts... the system sketching out the relationships is a good addition to the summary of my field.
- There were few rules that were useful to me whose associations were not expected but when thinking deeply, the associations were in fact strong and worth exploring further.
- One interesting rule was *Professor_name -> area_of_research.* This would help identify persons with similar interests.
- Some rules confirmed relationships between concepts I already knew. Others where quite unexpected and provided me with hints on keywords that I could use to search for more information.
- They are useful because they show my some concept relationship I have not pay much attention before. They let me know that maybe these concepts together can explain something and if I research more on them perhaps I will find some useful information.
- Often, there are rules I didn't think of in my research papers.
- Some relations which I generally fail to recognize as a relation, and usually assume them to exist are better verbalized here.
- Some of the rules are useful since I didn't think about the relationship in this way until I saw the rules presented. It helped me looking those concepts from a different angle.

The comments from participants about the useful rules can be summarized into the following categories:

- Introducing new concepts. The rules themselves present some new concepts to the participants in their research areas.
- Introducing new relationships. The rules present new relationships between concepts in the participants' research areas.
- Summarizing important relationships in an area. The summarization may not be totally new, but it is good to the participants.
- Assisting participants in recognizing overlooked relationships.
- Offering an opportunity for exploring interesting things introduced by the rules. Participants may not be able to find the use of the rule immediately, but further exploration will enable them to find more about rules.

6.6 Summary

This chapter discussed the data analysis methods and the results. The subjective ratings of rules novelty and usefulness collected from participants were treated as the true values of the rule novelty and usefulness. The proposed user-oriented novelty measure and other interestingness measures identified from the literature were calculated for all discovered rules. The performance of the measures in identifying novel and useful rules were evaluated by calculating the correlations between the measures and the true values (user ratings). The relative performance of different measures was compared using their correlations with the true values, and measures with higher correlations with the true values were considered having better performance – novelty prediction accuracy and usefulness indication power.

The results show that the user-oriented novelty measure has a high correlation with the subjective novelty ratings and the subjective usefulness ratings, which suggests

that it can help users find both novelty and useful rules. Not only does the user-oriented novelty measure outperform the WordNet novelty measure in predicting novelty rules but also it outperforms the WordNet novelty measure and eight objective measures in identifying useful rules. The regression analyses demonstrate that various user factors do not have significant influence on the difference between the performance of the interestingness measures.

The analysis of the post-evaluation questionnaire gives a deeper understanding of the participants' experience in the user study and their opinions about the discovered rules. Their comments suggest different ways the user-oriented novelty measure can help users find useful rules.

CHAPTER 7

SUMMARY AND CONCLUSIONS

This chapter will summarize the major findings of this study, discuss the results in terms of theoretical and practical implications, outline the contributions of this study, and discuss the limitations and possible future research directions.

7.1 Summary of Findings

The main objectives of this study are to:

1. Develop a user-oriented text mining framework in which the users' background knowledge is implicitly derived, and exploited in the text mining process to discover tailored knowledge for particular users.
2. Develop a user-oriented novelty measure that can be calculated using the background knowledge data structure to evaluate the interestingness of discovered association rules.
3. Evaluate the performance of the user-oriented novelty measure in terms of identifying interesting (novel and useful) association rules.

The study was driven by these objectives, and the findings are summarized as follows.

7.1.1 The User-Oriented Text Mining Framework

The proposed framework differs greatly from existing text mining methodologies in that it consists of an additional component, the background knowledge developer, in the whole mining process. The background knowledge developer can learn a user's background knowledge and interests from various sources, such as the articles read by the user and the web pages visited by the user. The developed background knowledge can then be applied in each step of the text mining process, e.g. retrieving relevant target documents, extracting meaningful document features, and evaluating the interestingness of discovered patterns.

In this study, the proposed framework was applied to the problem of discovering novel association rules from text, and the background knowledge was exploited to evaluate the novelty of association rules discovered from target documents. The results show that the framework is working appropriately, and it could be adopted in other text mining tasks as well.

7.1.2 The Background Knowledge Data Structure

For the problem of discovering novel association rules, the users' background knowledge was modeled as a keyword space. A concept hierarchy was developed to capture the semantic relationships between concepts. Keywords were extracted from the background document provided by users, and the POCA technique was used to develop the concept hierarchy.

This structure captured two types of relationships among keywords: co-occurrence and connectivity. Keywords were indexed as an inverted list, in which each node consists of a keyword and a list of documents containing this keyword (including its

frequency in each document). This inverted list made it convenient to calculate the co-occurrences of two keywords. Connectivity among keywords was captured by the concept hierarchy, and it was measured by considering the connections of two keywords with other common keywords.

This structure is proved effective in the study for calculating the user-oriented novelty measure. The key algorithms relying on this structure have also been developed. The background knowledge data structure could be used for calculating other measures as well, for example, the similarity of a document to a user's background, which could be useful in a recommendation system or a user information agent system.

7.1.3 The User-Oriented Novelty Measure

The user-oriented novelty measure is the essential part of this study. The goal of this measure is to estimate the novelty of an association rule by measuring the distance of the antecedent and the consequent of a rule, so it is a distance measure in nature. The distance reflects how novel (previously unknown) a rule is to a particular user.

The novelty measure is decomposed into two distance components: occurrence distance and connection distance. The former looks at the overlapping area of two keywords: the more they overlap, the less distant they are. The latter employs the connectivity information of keywords in the concept hierarchy, and measures the distance between two keywords using the length of the shortest path between them in the concept hierarchy. Given an association rule, its novelty will be solely determined by the user's background knowledge (modeled as a keyword space). Therefore, the novelty measure is called user-oriented novelty measure.

The user-oriented novelty measure is also proposed as an interestingness measure. It is assumed that novel association rules are also useful to users, because they convey something unknown to users. Users could either obtain new knowledge from these novel association rules directly, or start from these rules to learn related knowledge. In the later case, these novel rules could serve as triggers or incentives for users to learn new knowledge. From the study, it is found that the majority of the participants thought novel rules were also useful (these was a high correlation between the subjective novelty ratings and the subjective usefulness ratings of association rules), so, in most cases, identifying novel rules can lead to the finding of useful rules.

7.1.4 The Evaluation

The evaluation focused on studying the novelty prediction accuracy and the usefulness indication power of the user-oriented novelty measure. The questions that needed to be answered were (1) how effective the user-oriented novelty measure is in term of correlating with human judgments, and (2) how good the user-oriented novelty measure is as an interestingness measure when compared with other measures.

The main findings from the evaluation are summarized as follows:

1. The user-oriented novelty measure has high novelty prediction accuracy. Novelty prediction accuracy was measured by comparing the predicted measure scores to the true novelty values (the subjective novelty ratings). Compared to the results of prior studies, there is a high correlation (about 0.38) between the user-oriented novelty predictions and the true values.
2. The user-oriented novelty measure significantly outperforms the WordNet novelty, Support, and Confidence with respect to novelty prediction. Such a

significant phenomenon could be due to the user-oriented novelty measure having a higher correlation with the subjective novelty ratings than the WordNet novelty, Support and Confidence.

3. The user-oriented novelty measure has high usefulness indication power. Usefulness indication power was measured by comparing the predicted measure scores to the true usefulness values (the subjective usefulness ratings). In general, the correlation between the user-oriented novelty scores and the subjective usefulness ratings is about 0.17. There are four cases in the evaluation study in which the correlations between the subjective novelty ratings and the subjective usefulness ratings are negative. A negative correlation means that the participant does not think that novel rules are useful, so for those participants novelty measures will no longer be effective for finding useful rules. However, the majority of the participants still have positive correlations between their subjective novelty ratings and their subjective usefulness ratings, so for them it is still possible to use novelty measures to identify useful rules. After the four cases are removed from analysis, the correlation between the user-oriented novelty scores and the subjective usefulness ratings increases to 0.24. Therefore, it can be concluded that, in most cases, the user-oriented novelty measure has high usefulness indication power.
4. The user-oriented novelty measure outperforms the WordNet novelty and other seven objective interestingness measures with respect to usefulness indication. Correlations between other measures and the subjective usefulness ratings are also calculated. The comparison of the correlations of the user-oriented novelty

measure, the WordNet novelty, and other seven objective interestingness measures show that the user-oriented novelty measure has the highest correlation. The difference is significant for all objective measures. Although, in general, the difference is not significant for the WordNet measure, it becomes significant when the four cases mentioned above are removed. The conclusion is that, in term of finding useful association rules, the user-oriented novelty measure is significantly better than compared objective measures, and in most cases, it is significantly better than the WordNet novelty measure.

7.1.5 Answers to Research Questions

The answers to the research questions in Chapter 5 are summarized in Table 7.1.

Table 7.1 Answers to All Research Questions

Q1	Can the user-oriented novelty measure help users find interesting rules?
A	Yes
Q1.1	Is the novelty prediction accurate?
A	The user-oriented novelty prediction is highly correlated with human subjective novelty ratings (correlations: 0.38)
Q1.2	Are novel association rules also useful to users?
A	In most cases, yes. 21 out of 25 participants have high positive correlations between *SN* and *SU*, which suggests that novelty measures could be a good usefulness indicator.
Q2	Does the user-oriented novelty measure perform better than other interestingness measures?
A	Yes
Q2.1	Does the user-oriented novelty measure perform better than other novelty measures in predicting the novelty of association rules?
A	Yes. The user-oriented novelty measures perform significantly better than the WordNet novelty measure in predicting the novelty of association rules.
Q2.2	Does the user-oriented novelty measure perform better than other interestingness measures in terms of identifying useful association rules?
A	Yes. The user-oriented novelty measures perform significantly better than objective measures in identifying useful rules. It is more correlated with human subjective usefulness ratings than the WordNet novelty measure.

Table 7.1 Answers to All Research Questions (Continued)

Q3	What are the factors affecting the performance of the user-oriented novelty measure?
A	In this study, two types of factors were investigated: the number of background documents and user difference.
Q3.1	How many background documents are needed?
A	In this study, no effect of the number of background documents was found on the measure performance. The minimum number of background documents provided by one participant is 12, which suggests that the proposed method does not require a large number of background documents to generate good performance.
Q3.2	Do user differences have effects on the system performance?
A	In this study, no effect of user differences on the measure performance was found, which suggests that the system can work well for different types of users.

7.2 Theoretical and Practical Implications

The findings of this study contribute to both the literature of text mining and data mining, and the practices of knowledge discovery systems.

7.2.1 Theoretical Implications

Text mining is a relatively new research field, and it draws on several related fields, such as information retrieval, natural language processing, and information extraction. The proposed method for discovering novel association rules from documents successfully integrates several existing techniques in different research fields, such as noun phrase extraction in natural language processing, concept hierarchy development in information retrieval, association rules mining in data mining. The integration is not just putting everything together; instead it is achieved by carefully arranging all components within the user-oriented text mining framework.

The proposed user-oriented novelty measure is another contribution to the data mining literature. The interestingness problem is attracting more attention, and there are many interestingness measures proposed in the data mining literature to tackle this problem, including objective measures and subjective measures. Objective measures are easy to calculate, but not sufficient because of the lack of consideration of user's interests. Subjective measures are more effective, but require more user input. In a text mining environment, such required explicit user input is very hard to obtain. The user-oriented novelty measure has the advantages of both the objective and the subjective measures. On one hand, it can be calculated objectively once the use's background knowledge has been learned from the background documents. On the other hand, it takes into consideration of individual's knowledge, so that the results are subjectively customized for different users. The essential part of the user-oriented novelty measure is the usage of the subjective knowledge implicitly learned from a set of documents.

The proposed methodology provides users with an overview of what he/she has read and/or written and what is new in a new document set. The availability of the background knowledge and the recommended new knowledge may eventually change the way people search for information. Currently the most popular way of finding relevant information is using a search engine. This method requires users to know their information need, to be able to translate the information need into a query acceptable by a search engine, and to judge the returned documents to identify relevant ones. If a user initially has some knowledge about certain aspects of a topic, he/she is likely to use such knowledge to search for new information, which in turn gives him/her more information on those aspects. The user may have to read through a lot of documents in order to

obtain some new knowledge. This incremental knowledge acquisition process is slow and difficult for user to gain new knowledge. With the proposed system, it becomes easier for the user to know what is missing in his/her background knowledge, thus he/she could choose to search for information that is absent in his/her current knowledge. Therefore, the system could help users expand their knowledge rapidly and easily.

7.2.2 Practical Implications

First, the proposed user-oriented text mining framework and the methodology for mining novel association rules from text can be implemented as a fully functional personalized knowledge discovery system. The prototypes developed for this study and the evaluation results demonstrate the feasibility of the proposed method for personalized knowledge discovery from documents. An application environment can be created by integrating a research article management program, a feature extraction program, an association rule mining program, and the novelty evaluation program. The integrated application environment will provide better usability and user interactions for users to organize their writings and readings and discover useful knowledge from new documents.

Second, the background knowledge development method can be applied to build user profiles in many knowledge discovery systems. The background knowledge developed for this study contains not only the keywords in a user's background documents but also the relationships between keywords. It captures more semantic usage of keywords and their relationships than do most existing user profiles that are represented as bag of keywords. The concept hierarchy enables more advanced calculation for measuring the distance or similarity between keywords, which in turn could be used to evaluate and refine different types of discovered patterns, such as

decision rules, classification rules and topic detections. The concept hierarchy could be useful in many information management systems as well. For example, many search engines provide recommendations of related search keywords to the original query so that it is easier for the user to expand the original query or start over with a more relevant query. The recommendations can be generated by looking at the co-occurrences of keywords in the search results. When the concept hierarchy is available, the system will be able to generate different types of candidate keywords, such as more general or specific keywords (by moving up or down along a certain path in the hierarchy) and keywords leading to more familiar or unknown results (by calculating the novelty between the candidate keywords and the original search keywords).

Third, many recommendation systems can take advantage of the proposed algorithms as well. The recommendations could be new concepts or documents containing interesting concepts. They can be generated by simply analyzing the content of new documents or the opinions from people sharing similar interests. The content-analysis method is similar to what has described in this study. In the second situation, the background knowledge of all users can be developed and the similarity between every pair of users' profiles can be calculated, so that groups of users sharing similar interests can be identified. The system can then provide a user with recommendations of concepts and documents that are rated novel or useful by other users in the same group. Compared with the content analysis method, the second one is more related to the collaborative filtering technology. Better results could be obtained by combining the two methods together.

The results from this study provide practitioners with a few suggestions when developing personalized text mining system. (1) Background document collecting should be as automatic as possible, because it requires some effort for users to submit their background documents. In addition to the manual operation, the system may allow advanced operations that require less user effort, such as drag and drop and automatic recording of users' browsing history. (2) The system should be interactive. During the knowledge discovery process, users may want to control the process or the intermediate results. For example, the system should allow users to update the background knowledge, or adding to or removing from the feature list. (3) The system should have a good interface for presenting the discovered rules. In particular there will be tens of thousands of rules discovered by the system. It is extremely difficult for users to go through the list without a good presentation interface. The system could group rules according to the common terms they have, or provide a filtering function to help users eliminate rules that are of interests at a particular time. (4) The system should be efficient. In practice, a text mining system usually deals with a large number of documents. Some of the processes are time consuming. An efficient system will produce the results with fewer resources in less time, which requires efficient algorithms and a good system architecture. This study has presented the key algorithms and their VC++ implementations. There is still room for improvement and new algorithms.

7.3 Contributions

This study has made the following contributions to the data/text mining literature and the field of information systems.

1. Developed the user-oriented novelty measure to evaluate the interestingness of discovered association rules. In text mining tasks, most of the approaches take a data-oriented view, so the (objective) measures developed for interestingness evaluation are solely determined by the rules and the underlying data collection, without the consideration of users' background and interests. Though subjective measures consider users' interests, they require explicit input of users' expectations. Little work has been done to implicitly learn the users' background knowledge and exploit such knowledge in the text mining process. This study is by far the first attempt to develop a low user-effort but high user-orientedness interestingness measure for association rule mining from text. The results are promising. The proposed measure outperforms the WordNet novelty measure and seven (7) other objective measures for identifying novel and useful association rules from text.
2. Developed the key algorithms for calculating the user-oriented novelty measure, including the concept hierarchy development algorithm, the hierarchy shortest path search algorithm, and the hierarchy distance calculation algorithm.
3. Developed a user-oriented text mining framework. This study is one of the few studies that address the problem of text mining from the user's perspective. By taking account of the user's background knowledge, the system can identify novel

and useful patterns particularly for a user. A working prototype is implemented, which shows that the proposed methodology is practically feasible.

4. Evaluated the proposed user-oriented novelty measure. In the evaluation, the task, participant recruitment, and study procedure were carefully planned. A set of tools, including the article uploading tool and the online evaluation system, were developed. The evaluation results provide a better understanding of the proposed methodology and the affecting factors on the system performance.

7.4 Future Directions

Because of the time limit on this study, a few assumptions had to be made (see Chapter 3 for details), and they may prevent the findings from being generalized to other situations. In the near future, the study can be expanded in the following directions:

1. Consider the dynamic changes of users' background knowledge. In this study, users' background knowledge is assumed static after the participants submit their background documents into the system. This assumption is fine with one-time association rule mining and novelty calculations. If the user chooses to use the system for a period of time, it is necessary to consider the changes of the user's background knowledge which is reflected by the updates of the background documents. When new background documents are added to the system, the existing background knowledge should be updated dynamically, and the novelty scores of all affected association rules should be recalculated. The system may also give higher preferences to more recent documents, so that the background knowledge will reflect the user's most current interests.

2. Expand the framework to group level. In this study, the background knowledge is developed for individual users. It is also very common that members in a group share similar interests and have expertise in different specific areas. It is desirable for them to know what expertise other members have could benefit their own work. In this case, all group members could contribute their background documents to the system, and the system could then develop a group profile (background knowledge) with the track of the sources of items in the background knowledge. Therefore, the system could provide each member with information about the knowledge of other members. In addition, it could discover novel and useful knowledge from new documents for the entire group.
3. Consider the asymmetry of the user-oriented novelty measure. In this study, the user-oriented novelty measure is symmetric. In other words, the novelty of *A* -> *B* is the same as the novelty of *B* -> *A*. This may not reflect the asymmetric nature of association rules. The above two rules describe two different associations between *A* and *B,* if they have different confidence scores. In the future work, the direction between concepts in association rules needs to be considered. For example, the hierarchy distance could be calculated as the sum of the weights, rather than the number of edges in the shortest path as the distance between two concepts. The weight of an edge could be the confidence score between the starting concept and the ending concept.
4. Provide local context of an association rule for easier comprehension and better usability. One of the feedbacks from the current study is that it is very difficult to tell whether a rule is useful without looking at the context where the concepts

occur. Therefore, one question for the future research is how the local context can be added to the system, and whether adding the local context of association rules help users understand them.

5. Enhance the prototype system and test it with end users. In this study, the uMining system was run by the researcher, and only the results were presented to the participants for evaluation. In the future work, user evaluations with the uMining system need to be conducted to better understanding the usability and perceived usefulness of the system.
6. Conduct broader user studies. The current user study was conducted with Ph.D. students, and the documents were limited to their research interests. The future user studies will recruit participants from different domains at different levels, and design more tasks. The broader user studies will reveal whether the proposed methodology works for different types of users and tasks.

7.5 Summary

This chapter concludes the dissertation with the main findings, the major contributions, and the possible future directions of the study.

APPENDIX A

STOP WORD LIST

Appendix A contains the list of stop words used in this study.

a
about
above
across
after
afterwards
again
against
all
almost
alone
along
already
also
although
always
am
among
amongst
amoungst
amount
amp
an
and
another
any
anyhow
anyone
anything
anyway
anywhere
are
around
as
at
back
be
became
because
become
becomes
becoming
been
before
beforehand
behind
being
below
beside
besides
between
beyond
bill
both
bottom
but
by
call
can
cannot
cant
co
con
copyright
could
couldnt
cry
de
describe
despite
detail
do
done
down
due
during
each
eg
eight
either
eleven
else
elsewhere
empty
enough
etc
even
ever
every
everyone
everything
everywhere
except
few
fifteen
fify
fill
find
fire
first
five
for
former
formerly
forty
found
four
from
front
full
further
get
give
go
had
has
hasnt
have
he
hence
her
here
hereafter
hereby
herein
hereupon
hers
herself
him
himself
his
home
homepage
how
however
hundred
i
ie
if
in
inc
indeed
interest
into
is
it
its
itself
keep
last
latter
latterly
least
less
ltd
made

many
may
me
meanwhile
might
mill
mine
more
moreover
most
mostly
move
much
must
my
myself
name
namely
neither
never
nevertheless
next
nine
no
nobody
none
noone
nor
not
nothing
now
nowhere
of
off
often
on
once
one
only
onto
or
other
others
otherwise
our
ours
ourselves
out
over
own
page
part
per
perhaps
please
put
rather
re
right
same
see
seem
seemed
seeming
seems
serious
several
she
should
show
side
since
sincere
site
six
sixty
so
some
somehow
someone
something
sometime
sometimes
somewhere
still
such
system
take
ten
than
that
the
their
them
themselves
then
thence
there
thereafter
thereby
therefore
therein
thereupon
these
they
thick
thin
third
this
those
though
three
through
throughout
thru
thus
to
together
too
top
toward
towards
twelve
twenty
two
un
under
until
up
upon
us
very
via
was
we
web
well
were
what
whatever
when
whence
whenever
where
whereafter
whereas
whereby
wherein
whereupon
wherever
whether
which
while
whither
who
whoever
whole
whom
whose
why
will
with
within
without
would
yet
you
your
yours
yourself
yourselves

APPENDIX B

CONSENT FORM

Appendix B contains the consent form used in the user study.

NEW JERSEY INSTITUTE OF TECHNOLOGY
323 MARTIN LUTHER KING BLVD.
NEWARK, NJ 07102

CONSENT TO PARTICIPATE IN A RESEARCH STUDY

TITLE OF STUDY: Text Mining with the Exploitation of User's Background Knowledge

RESEARCH STUDY:

I, _____, have been asked to participate in a research study under the direction of Xin Chen. Other professional persons who work with them as study staff may assist to act for them.

PURPOSE: To evaluate the effectiveness of a data mining algorithm for knowledge discovery from text.

DURATION:

My participation in this study will last for __up to 6 weeks_____.

PROCEDURES:

I have been told that, during the course of this study, the following will occur:
1. I will be asked to voluntarily use an online system to manage my personal research articles.
2. I will be asked to voluntarily complete one online survey after I have used the system for a while.

PARTICIPANTS:

I will be one of about ____50____participants to participate in this trial.

EXCLUSIONS:

I will inform the researcher if any of the following apply to me:
- I do not wish to use the system for any reason.

- I do not wish to complete the survey for any reason.

RISK/DISCOMFORTS:

I have been told that the study described above may involve the following risks and/or discomforts:
- None known or anticipated discomforts. Security of the system might be at risk of computer hacking, as it is in any computer system. Every effort (e.g. blocking the unused ports, update the system with the latest patches, and checking system logs as frequently as possible to catch abnormal usage) will be made to keep the system secure from hacking.

There also may be risks and discomforts that are not yet known.

I fully recognize that there are risks that I may be exposed to by volunteering in this study which are inherent in participating in any study; I understand that I am not covered by NJIT's insurance policy for any injury or loss I might sustain in the course of participating in the study.

CONFIDENTIALITY:

Every effort will be made to maintain the confidentiality of my study records. Officials of NJIT will be allowed to inspect sections of my research records related to this study. If the findings from the study are published, I will not be identified by name. My identity will remain confidential unless disclosure is required by law.

PAYMENT FOR PARTICIPATION:

I have been told that I will receive no monetary compensation for my participation in this study. However, free access to an online research article management system will be awarded for my participation in the study.

RIGHT TO REFUSE OR WITHDRAW:

I understand that my participation is voluntary and I may refuse to participate, or may discontinue my participation at any time with no adverse consequence. I also understand that the investigator has the right to withdraw me from the study at any time.

INDIVIDUAL TO CONTACT:

If I have any questions about my treatment or research procedures that I discuss them with the principal investigator. If I have any addition questions about my rights as a research subject, I may contact:

Dawn Hall Apgar, PhD Chair, IRB (973) 642-7616

SIGNATURE OF PARTICIPANT

I have read this entire form, or it has been read to me, and I understand it completely. All of my questions regarding this form or this study have been answered to my complete satisfaction. I agree to participate in this research study.

Subject Name:_____
Signature:_____

Date:_____

SIGNATURE OF READER/TRANSLATOR IF THE PARTICIPANT DOES NOT READ ENGLISH WELL

The person who has signed above,
_____, does not read English well, I read English well and am fluent in (name of the language)
_____, a language the subject understands well. I have translated for the subject the entire content of this form. To the best of my knowledge, the participant understands the content of this form and has had an opportunity to ask questions regarding the consent form and the study, and these questions have been answered to the complete satisfaction of the participant (his/her parent/legal guardian).

Reader/Translator Name:_____
Signature:_____
Date:_____

SIGNATURE OF INVESTIGATOR OR RESPONSIBLE INDIVIDUAL

To the best of my knowledge, the participant,
_____, has understood the entire content of the above consent form, and comprehends the study. The participants and those of his/her parent/legal guardian have been accurately answered to his/her/their complete satisfaction.

Investigator's Name:_____
Signature:_____

Date:_____

APPENDIX C

BACKGROUND QUESTIONNAIRE

Appendix C contains the background question used in the user study.

Dear _____,

Before participating in the study, please take a few minutes to give us some background information of you and your research. The information you provide will help us achieve a better understanding of the evaluation results. Do not worry about projecting a good image. Your answers are strictly confidential and will NOT affect your usage of the system in anyway.

Gender: [ ] Male [ ] Female [ ] Keep Confidential

Is English your native or first language? [ ] Yes [ ] No

Major: _____ Specialization: _____

Current stage of my research:
[ ] Looking for a topic
[ ] Completing SOTA (State-of-the-Art Literature Review)
[ ] Completing proposal
[ ] Completing dissertation
[ ] Finished PhD study
[ ] Other (please specify) _____

I have ___ years of studying/working experience in Information Systems or related fields.

My research interests:

_____

_____

On average, how many hours per day do you spend on computer to complete your work?
[ ] Less than 2 [ ] 2~4 [ ] 4~6 [ ] 6~8 [ ] More than 8

Have you heard of Text Mining before? [ ] Yes [ ] No
If yes, have you ever used any Text Mining systems? [ ] Yes [ ] No

Thank you,

Xin Chen

APPENDIX D

EVALUATION INSTRUCTIONS

Appendix D contains the evaluation instructions used in the user study.

What is an association rule?

An example of such rule is Bread, Butter --> Milk, which describes the fact that many of the customers who purchased Bread and Butter also purchased Milk. In this evaluation, the rules that you are going to evaluate are extracted from research papers that are related to your research interests. They describe how concepts are associated in these areas.

How should I evaluate a rule?

To evaluate a rule, please choose the most appropriate choice for each measure. Plausibility means the established relationship sounds reasonable, novelty is the extent to which the relationship is new to you within the context of your research interests, and usefulness is to what degree you think the relationships help you acquire new knowledge or understand existing concepts in your research areas. Please choose the most appropriate choice from the answers (1 for the least and 7 for the most).

If there is any question or problem, please do not hesitate to contact me.

Thank you.

APPENDIX E

RULE EVALUATION

Appendix E describes the sample rule format and evaluation scales.

Please choose the most appropriate response to each question on a 7-point scale (1 for the least and 7 for the most).

1. [Wanda] -> [business]
   Plausibility [ ]1 [ ]2 [ ]3 [ ]4 [ ]5 [ ]6 [ ]7 [ ]Can't tell
   Novelty [ ]1 [ ]2 [ ]3 [ ]4 [ ]5 [ ]6 [ ]7 [ ]Can't tell
   Usefulness [ ]1 [ ]2 [ ]3 [ ]4 [ ]5 [ ]6 [ ]7 [ ]Can't tell

2. [Info] -> [Contact College]
   Plausibility [ ]1 [ ]2 [ ]3 [ ]4 [ ]5 [ ]6 [ ]7 [ ]Can't tell
   Novelty [ ]1 [ ]2 [ ]3 [ ]4 [ ]5 [ ]6 [ ]7 [ ]Can't tell
   Usefulness [ ]1 [ ]2 [ ]3 [ ]4 [ ]5 [ ]6 [ ]7 [ ]Can't tell

3. [Thomas Jeneklasen] -> [Photography]
   Plausibility [ ]1 [ ]2 [ ]3 [ ]4 [ ]5 [ ]6 [ ]7 [ ]Can't tell
   Novelty [ ]1 [ ]2 [ ]3 [ ]4 [ ]5 [ ]6 [ ]7 [ ]Can't tell
   Usefulness [ ]1 [ ]2 [ ]3 [ ]4 [ ]5 [ ]6 [ ]7 [ ]Can't tell

4. [formats] -> [Sample Chapter]
   Plausibility [ ]1 [ ]2 [ ]3 [ ]4 [ ]5 [ ]6 [ ]7 [ ]Can't tell
   Novelty [ ]1 [ ]2 [ ]3 [ ]4 [ ]5 [ ]6 [ ]7 [ ]Can't tell
   Usefulness [ ]1 [ ]2 [ ]3 [ ]4 [ ]5 [ ]6 [ ]7 [ ]Can't tell

5. [navigation menu] -> [Summer]
   Plausibility [ ]1 [ ]2 [ ]3 [ ]4 [ ]5 [ ]6 [ ]7 [ ]Can't tell
   Novelty [ ]1 [ ]2 [ ]3 [ ]4 [ ]5 [ ]6 [ ]7 [ ]Can't tell
   Usefulness [ ]1 [ ]2 [ ]3 [ ]4 [ ]5 [ ]6 [ ]7 [ ]Can't tell

6. [forms] -> [Student Link]
   Plausibility [ ]1 [ ]2 [ ]3 [ ]4 [ ]5 [ ]6 [ ]7 [ ]Can't tell
   Novelty [ ]1 [ ]2 [ ]3 [ ]4 [ ]5 [ ]6 [ ]7 [ ]Can't tell
   Usefulness [ ]1 [ ]2 [ ]3 [ ]4 [ ]5 [ ]6 [ ]7 [ ]Can't tell

Note: This sample shows the format and evaluation scales that were used in the user study. For each participant, the rules are different.

APPENDIX F

POST-EVALUATION QUESTIONNAIRE

Appendix F contains the post-evaluation questionnaire used in the user study.

Thank you for participating in my study. Please answer the following questions regarding the study and your experiences. Fill the answer that best fits your immediate reaction. Do not worry about projecting a good image. Your answers are strictly confidential.

1. I consider myself _____ in the area related to my research interests.

   Novice [1] [2] [3] [4] [5] [6] [7] Expert
2. The articles I provided represent my knowledge about the field related to my research interests.

   Strongly disagree [1] [2] [3] [4] [5] [6] [7] Strongly agree
3. The rules presented for evaluation are related to my research interests.

   Strongly disagree [1] [2] [3] [4] [5] [6] [7] Strongly agree
4. There are rules which introduce me new concepts related to my research interests.

   Strongly disagree [1] [2] [3] [4] [5] [6] [7] Strongly agree
5. There are rules which help me identify new relationships between concepts related to my research interests.

   Strongly disagree [1] [2] [3] [4] [5] [6] [7] Strongly agree
6. Knowing these rules DOES NOT help my research.

   Strongly disagree [1] [2] [3] [4] [5] [6] [7] Strongly agree
7. A system with the capability to extract association rules is useful.

Strongly disagree [1] [2] [3] [4] [5] [6] [7] Strongly agree

8. Identifying relationships among concepts in my research area is NOT important to me.

   Strongly disagree [1] [2] [3] [4] [5] [6] [7] Strongly agree

9. The instructions are clear and I understood the procedure of the study.

   Strongly disagree [1] [2] [3] [4] [5] [6] [7] Strongly agree

10. I had difficulties with the online system during the rule evaluation.

    Strongly disagree [1] [2] [3] [4] [5] [6] [7] Strongly agree

11. Did you find useful rules from the evaluation set? If yes, please briefly describe why and how you think they are useful.

12. What did you like best about the system?

13. What do you think should be improved to the system?

14. Do you have any other comments about the study and the system?

APPENDIX G

VC++ SOURCE CODE

This appendix contains the VC++ source code of the key algorithms in Chapter 4.

G.1 Concept Hierarchy Development

```
UINT CHierarchyDeveloper::DevelopHierarchy(LPVOID pParam)
{
	CHierarchyDeveloper* pDeveloper = (CHierarchyDeveloper*)pParam;

	pDeveloper->m_bStop = FALSE;

	//remove terms with less df or excluded
	CArray<CTerm*, CTerm*> aryTerms;
	for(int i=0; i< pDeveloper->m_pTermArray->GetTermCount(); i++)
	{
		CTerm* pTerm = pDeveloper->m_pTermArray->GetTerm(i);
		if(pTerm->IsFlag() &&
			pTerm->m_docList.GetCount() >= *pDeveloper->m_pDocFreq)
		{
			aryTerms.Add(pTerm);
		}
	}

	//sort terms by df
	CString strMess = "Sorting features by document frequency...";
	SortTerms(aryTerms);

	//build hierarchy -- up down
	pDeveloper->m_pParent->SendMessage(ID_UMINING_NOTIFY, ID_TOTAL, iCount);
	for(i=0; i<iCount && !pDeveloper->m_bStop; i++)
	{
		CTerm *pTerm = aryTerms.GetAt(i);

		CList<CTerm*, CTerm*> lstParents;
		pDeveloper->AddTermToTree(pDeveloper->m_pHierarchy,
			pTerm, *pDeveloper->m_pThreshold, lstParents);
		lstParents.RemoveAll();

		pDeveloper->m_pParent->SendMessage(ID_UMINING_NOTIFY, ID_FINISH_ONE);
	}

	//find the depth of the hierarchy
	pDeveloper->m_pParent->SendMessage(ID_UMINING_NOTIFY, ID_TOTAL, iCount);
	int iMaxDepth = 0;
	for(i=0; i<iCount && !pDeveloper->m_bStop; i++)
	{
		CTerm *pTerm = aryTerms.GetAt(i);
		if(pTerm->m_aryChildren.GetSize() == 0)
		{
			if(pTerm->m_aryParent.GetSize() == 0 &&
				pTerm->m_aryAlias.GetSize() > 0)
```

```
                                        pTerm = pTerm->m_aryAlias[0];

                                int iDepth = pDeveloper->GetMaxDistanceToRoot(pTerm);
                                if(iDepth > iMaxDepth) iMaxDepth = iDepth;
                        }

                        pDeveloper->m_pParent->SendMessage(ID_UMINING_NOTIFY, ID_FINISH_ONE);
                }
                *pDeveloper->m_pMaxDepth = iMaxDepth;

                //display it
                strMess = "Displaying concept hierarchy...";
                pDeveloper->m_pParent->SendMessage(ID_UMINING_NOTIFY, ID_NOTIFICATION,
                        (LPARAM)(LPCTSTR)strMess);
                pDeveloper->m_pParent->SendMessage(ID_UMINING_NOTIFY, ID_FINISH_ALL);

                return ERR_SUCCESS;
}

//this algorithm works only when the terms are sorted
//by their document frequency in descending order
//return: 0: not added, 1: added as a child
UINT CHierarchyDeveloper::AddTermToTree(CTerm* pTree, CTerm *pTerm,
                                double dblThreshold, CList<CTerm*, CTerm*>& lstParents)
{
        if(lstParents.Find(pTree)) return 1;

        double dblPxy = pTree->GetCoOccurrenceProb(pTerm,
                m_pDocList->GetCount());
        double dblPyx = pTerm->GetCoOccurrenceProb(pTree,
                m_pDocList->GetCount());

        if(dblPxy > dblPyx && dblPxy >= dblThreshold)
        {
                UINT nRet = 0;
                for(int k=0; k<pTree->m_aryChildren.GetSize() && !m_bStop; k++)
                {
                        lstParents.AddTail(pTree);
                        CTerm* pChild = pTree->m_aryChildren.GetAt(k);
                        UINT nRet1 = AddTermToTree(pChild, pTerm, dblThreshold, lstParents);
                        if(nRet1 == 0) lstParents.RemoveTail();
                        nRet |= nRet1;
                }

                //if it is not added to any child, add it here
                if(nRet == 0)
                {
                        pTree->AddChild(pTerm);
                        pTerm->AddParent(pTree);
                        lstParents.AddTail(pTree);

                        return 1;
                }
                else
                {
                        return nRet;
                }
        }

        return 0;
}
```

G.2 Shortest Path Searching

```
int CHierarchyDeveloper::GetHierarchyDistance2(CTerm *pTerm1,
                        CTerm *pTerm2, int &nCurr, int &nShortest)
{
        if(nCurr >= nShortest || pTerm1 == pTerm2)
        {
                if(nCurr < nShortest) nShortest = nCurr;
                return nCurr;
        }

        nCurr++;
        int i;
        for(i=0; i<pTerm1->m_aryChildren.GetSize(); i++)
        {
                if(nCurr >= pTerm1->m_aryChildren[i]->m_nBestSoFar)
                        continue;

                pTerm1->m_aryChildren[i]->m_nBestSoFar = nCurr;

                GetHierarchyDistance2(pTerm1->m_aryChildren[i], pTerm2, nCurr, nShortest);
        }
        for(i=0; i<pTerm1->m_aryParent.GetSize(); i++)
        {
                if(nCurr >= pTerm1->m_aryParent[i]->m_nBestSoFar)
                        continue;

                pTerm1->m_aryParent[i]->m_nBestSoFar = nCurr;

                GetHierarchyDistance2(pTerm1->m_aryParent[i], pTerm2, nCurr, nShortest);
        }
        nCurr--;

        return nShortest;
}
```

G.3 Hierarchy Distance Calculation

```
double CHierarchyDeveloper::GetHierarchyDistance(
                CString strTerm1, CString strTerm2)
{
        double dblDistance = 0.0;

        if(strTerm1.Compare(strTerm2) == 0)
                return 0.0;

        int iPos;
        CTerm* pTerm1 = m_pTermArray->FindTerm(strTerm1, iPos);
        CTerm* pTerm2 = m_pTermArray->FindTerm(strTerm2, iPos);

        if(pTerm1 == NULL && pTerm2 == NULL)
        {
                //both are not found in the background knowledge
                dblDistance = 2 * (H + 2);
        }
        else if(pTerm1 == NULL && pTerm2->m_aryParent.GetSize() == 0)
        {
                //only one in background, but not in the hierarchy
                dblDistance = (H+2) + (H+1);
        }
```

```
        else if(pTerm2 == NULL && pTerm1->m_aryParent.GetSize() == 0)
        {
                //only one in background, but not in the hierarchy
                dblDistance = (H+1) + (H+2);
        }
        else if(pTerm1 == NULL && pTerm2->m_aryParent.GetSize() > 0)
        {
                //only one in background, and in the hierarchy
                dblDistance = (H+2) + GetMinDistanceToRoot(pTerm2);
        }
        else if(pTerm2 == NULL && pTerm1->m_aryParent.GetSize() > 0)
        {
                //only one in background, and in the hierarchy
                dblDistance = GetMinDistanceToRoot(pTerm1) + (H+2);
        }
        else if(pTerm1->m_aryParent.GetSize()+pTerm2->m_aryParent.GetSize() == 0)
        {
                //both in background, but none in hierarchy
                dblDistance = 2 * (H+1);
        }
        else if(pTerm1->m_aryParent.GetSize() == 0)
        {
                //both in background, but only one in hierarchy
                dblDistance = (H+1) + GetMinDistanceToRoot(pTerm2);
        }
        else if(pTerm2->m_aryParent.GetSize() == 0)
        {
                //both in background, but only one in hierarchy
                dblDistance = GetMinDistanceToRoot(pTerm1) + (H+1);
        }
        else
        {
                //both in hierarchy
                for(int i=0; i<m_pTermArray->GetTermCount(); i++)
                        m_pTermArray->GetTerm(i)->m_nBestSoFar = 999999;
                int iCurr = 0, iShortest = 999999;
                GetHierarchyDistance2(pTerm1, pTerm2, iCurr, iShortest);
                dblDistance = iShortest;
        }

        return dblDistance/(2*(H+2));
}

int CHierarchyDeveloper::GetMinDistanceToRoot(CTerm *pTerm)
{
        CArray<CInvertedList, CInvertedList&> aryPathList;
        CInvertedList ilPath;
        ilPath.AddTail(pTerm);
        GetAllPathToRoot(aryPathList, ilPath);

        //find the minimum distance
        int iMin = 999999;
        for(int m=0; m<aryPathList.GetSize(); m++)
        {
                if(aryPathList[m].GetCount() < iMin)
                        iMin = aryPathList[m].GetCount();
        }

        return iMin;
}
```

G.4 Occurrence Distance Calculation

```
double CHierarchyDeveloper::GetOccuDis(CString strTerm1, CString strTerm2)
{
	double dblSimi = 0.0;

	CStringArray saTerm1, saTerm2;
	strTerm1 += ' ';
	strTerm2 += ' ';

	int iBegin = 0, iEnd = 0;
	while( (iEnd = strTerm1.Find(' ', iBegin)) > 0)
	{
		CString strTerm = strTerm1.Mid(iBegin, iEnd-iBegin);
		if(!strTerm.IsEmpty()) saTerm1.Add(strTerm);

		iBegin = iEnd + 1;
	}

	iBegin = iEnd = 0;

	while( (iEnd = strTerm2.Find(' ', iBegin)) > 0)
	{
		CString strTerm = strTerm2.Mid(iBegin, iEnd-iBegin);
		if(!strTerm.IsEmpty()) saTerm2.Add(strTerm);

		iBegin = iEnd + 1;
	}

	int iCount = 0, iPos = 0;
	for(int i=0; i<saTerm1.GetSize() && !m_bStop; i++)
	{
		CString strSub1 = saTerm1[i];

		CTerm* pTerm1 = m_pTermArray->FindTerm(strSub1, iPos);

		for(int j=0; j<saTerm2.GetSize() && !m_bStop; j++)
		{
			CString strSub2 = saTerm2[j];

			CTerm* pTerm2 = m_pTermArray->FindTerm(strSub2, iPos);

			if(pTerm1 != NULL && pTerm2 != NULL)
			{
				int iCommon = pTerm1->GetCommonDocNum(pTerm2);
				int iNum1 = pTerm1->m_docList.GetCount();
				int iNum2 = pTerm2->m_docList.GetCount();

				dblSimi += ((double)iCommon/min(iNum1, iNum2));;
			}

			iCount++;

			CProgressWnd::PeekPumpMessage();
		}
	}

	dblSimi /= iCount;

	return (1-dblSimi);
}
```

G.5 Connection Distance Calculation

```
double CHierarchyDeveloper::GetConnDis(CString strTerm1, CString strTerm2)
{
	double dblDistance = 0.0;

	CStringArray saTerm1, saTerm2;
	strTerm1 += ' ';
	strTerm2 += ' ';

	int iBegin = 0, iEnd = 0;
	while( (iEnd = strTerm1.Find(' ', iBegin)) > 0)
	{
		CString strTerm = strTerm1.Mid(iBegin, iEnd-iBegin);
		if(!strTerm.IsEmpty()) saTerm1.Add(strTerm);

		iBegin = iEnd + 1;
	}

	iBegin = iEnd = 0;

	while( (iEnd = strTerm2.Find(' ', iBegin)) > 0)
	{
		CString strTerm = strTerm2.Mid(iBegin, iEnd-iBegin);
		if(!strTerm.IsEmpty()) saTerm2.Add(strTerm);

		iBegin = iEnd + 1;
	}

	int iCount = 0;
	for(int i=0; i<saTerm1.GetSize() && !m_bStop; i++)
	{
		CString strSub1 = saTerm1[i];
		if(strSub1.CompareNoCase("of") == 0)
			continue;

		for(int j=0; j<saTerm2.GetSize() && !m_bStop; j++)
		{
			CString strSub2 = saTerm2[j];
			if(strSub2.CompareNoCase("of") == 0)
				continue;

			if(strSub1.CompareNoCase(strSub2) != 0)
				dblDistance += GetHierarchyDistance(strSub1, strSub2);

			iCount++;

			CProgressWnd::PeekPumpMessage();
		}
	}

	dblDistance /= iCount;

	return dblDistance;
}
```

REFERENCES

Agrawal, R., Imilienski, T. & Swami, A. (1993). Mining Association Rules Between Sets of Items in Large Datasets. *Proceedings of the ACM SIGMOD International Conference on the Management of Data*, pp. 207-216.

Agrawal, R. & Srikant, R. (1994). Fast Algorithms for Mining Association Rules. *Proceedings of the 20th International Conference on Very Large Data Bases,* pp. 487-499.

Agrawal, R., Mannila, H., Srikant, R., Toivonen, H. & Verkamo, A. I. (1996). Fast Discovery of Association Rules. In U. Fayyad et al. (Eds.), *Advances in Knowledge Discovery and Data Mining* (pp. 307-328). AAAI Press.

Ahonen-Myka, H., Heinonen, O., Klemettinen, M. & Verkamo, A. I. (1999). Finding Co-occurring Text Phrases by Combining Sequence and Frequent Set Discovery. *Proceedings of 16th International Joint Conference on Artificial Intelligence IJCAI-99 Workshop on Text Mining: Foundations, Techniques and Applications*, pp. 1-9.

Ahonen-Myka, H. (2002). Discovery of Frequent Word Sequences in Text. *The ESF Exploratory Workshop on Pattern Detection and Discovery in Data Mining*, Imperial College, London.

Barker, K. & Cornacchia, N. (2000). Using Noun Phrase Heads to Extract Document Keyphrases. *Proceedings of the Thirteenth Canadian Conference on Artificial Intelligence,* Montreal, Canada, pp. 40-52.

Basu, S., Mooney, R. J., Pasupuleti, K. V. & Ghosh, J. (2001). Evaluating the Novelty of Text-Mined Rules Using Lexical Knowledge. *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* San Francisco, CA, pp. 233-238.

Basu, S., Mooney, R. J., Pasupuleti, K. V. & Ghosh, J. (2001). Using Lexical Knowlege to Evaluate the Novelty of Rules Mined from Text. *Proceedings of NAACL 2001 Workshop on WordNet and Other Lexical Resources: Applications, Extensions and Customizations*. Pittsburg, PA, pp. 144-149.

Beil, F., Ester, M. & Xu, X. (2002). Frequent Term-based Text Clustering. *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Alberta, Canada, pp. 436-442.

Brill, E. (1992). A Simple Rule-Based Part of Speech Tagger. *Proceedings of ANLP-92 the 3rd Conference on Applied Natural Language Processing,* Trento, Italy, pp. 152-155.

Brill, E. (1995). Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part of Speech Tagging. *Computational Linguistics* 21, 543-565.

Brin, S., Motwani, R., Ullman, J.D. & Tsur, S. (1997). Dynamic Itemset Counting and Implication Rules for Market Basket Data. *Proceedings of the ACM SIGMOD International Conference on Management of Data,* pp.255-264.

Chen, K. & Chen, H. (1994). Extracting Noun Phrases from Large-Scale Texts: A Hybrid Approach and Its Automatic Evaluation. *Proceedings of the 32nd Annual Meeting of ACL*, New Mexico, pp. 234-241.

Church, K. W. (1988). A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. *Proceedings of the Second Conference on Applied Natural Language Processing*, pp. 136-143.

Cutting, D., Kupiec, J., Pedersen, J. & Sibun, P. (1992). A Practical Part-Of-Speech Tagger. *Proceedings of the Third Conference on Applied Natural Language Processing,* Trento, Italy, pp. 133-140.

Dermatas, E. & Kokkinakis, G. (1995). Automatic stochastic tagging of natural language texts. *Computational Linguistics* 21(2), 137-163.

DeRose, S. J. (1998). Grammatical Category Disambiguation by Statistical Optimization. *Computational Linguistics* 14(1), 31-39.

Dumais, S. (1991). Improving the Retrieval of Information from External Sources. *Behavior Research Methods, Instruments, & Computers* 23, 229-236.

English, J., Hearst, M., Sinha, R., Swearingen, K. & Yee, K. P. (2002). Flexible Search and Navigation Using Faceted Metadata. Retrieved July 2004, from http://bailando.sims.berkeley.edu/papers/flamenco02.pdf.

Fayyad, U., Piatetsky-Shapiro, G. & Smyth, P. (1996). From Data Mining to Knowledge Discovery: An Overview. In U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy (Eds.). *Advances in Knowledge Discovery and Data Mining (pp. 1-36).* MA: MIT Press.

Feldman, R. & Math, I. D. (1995). Knowledge Discovery in Textual Databases (KDT). *Proceedings of the First International Conference on Knowledge Discovery*, Menlo Park, CA, pp. 112-117.

Feldman, R. (1995). The FACT System. Retrieved July 2003, from http://www.cs.biu.ac.il/~feldman/fact.html.

Feldman, R., Dagan, I. & Hirsh, H. (1998). Mining Text Using Keyword Distributions. *Journal of Intelligent Information Systems: Integrating Artificial Intelligence and Database Technologies* 10(3), 281-300.

Feldman, R. & Hirsh, H. (1998). Exploiting Background Information in Knowledge Discovery from Text. *Journal of Intelligent Information Systems* 9(1), pp. 83-97.

Feldman, R., Aumann, Y., Libetzon, Y., Ankori, K., Schler, J. & Rosenfeld, B. (2001). A Domain Independent Environment for Creating Information Extraction Modules. *Proceedings of CIKM 2001*, pp. 586-588.

Fellbaum, C. D. (1998). WordNet: An Electronic Lexical Database. *MIT Press*, Cambridge, MA.

Forsyth, R. & Rada, R. (1986). *Adding an Edge in Machine Learning: Applications in Expert Systems and Information Retrieval* (pp. 198-212), Ellis Horwood Ltd.

Frank, E., Paynter, G., Witten, I., Gutwin, C. & Nevill-Manning, C. (1999). Domain-Specific Keyphrase Extraction. *Proceeding of the Sixteenth International Joint Conference On Artificial Intelligence*, San Mateo, CA, pp. 668-673.

Frawely, W. J., Piatesky-Shapiro, G. & Matheus, C. J. (1991). Knowledge Discovery in Databases: An Overview. In G. Piatetsky-Shapiro and W.J Frawley (Eds.). *Knowledge Discovery in Databases* (pp. 1-27). AAAI/MIT Press.

Fu, Y. & Han, J. (1995). Meta-rule-guided Mining of Association Rules in Relational Database. *Proceedings of 1995 International Workshop on Knowledge Discovery and Deductive and Object-Oriented Database*, Singapore, pp. 39-46.

Fung, B. C. M., Wang, K. & Ester, M. (2003). Hierarchical Document Clustering Using Frequent Itemsets. *Proceedings of the 2003 SIAM International Conference on Data Mining*, San Francisco, CA, pp. 59-70.

Grishman, R. (1997). Information Extraction Techniques and Challenges. In M. Pazienza (Ed.). *Information Extraction*. Springer-Verlag, Lecture Notes in Artificial Intelligence, Rome.

Han, J. & Fu, Y. (1994). Dynamic Generation and Refinement of Concept Hierarchies for Knowledge Discovery in Databases. *Proceedings of AAAI'94 Workshop on Knowledge Discovery in Databases*, Seattle, WA, pp. 157-168.

Han, J. & Fu, Y. (1999). Discovery of Multiple-Level Association Rules from Large Databases. *IEEE Transactions on Knowledge and Data Engineering* 11(5), 420-431.

Hearst, M. A. (1999). Untangling Text Data Mining. *Proceedings of ACL'99: the 37th Annual Meeting of the Association for Computational Linguistics*, University of Maryland (invited paper).

Hilderman, R. J. & Hamilton, H. J. (2001). Evaluation of Interestingness Measures for Ranking Discovered Knowledge. *Proceedings of the 5th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Hong Kong, pp. 247-259.

Jones, K. S. (1972). A Statistical Interpretation of Term Specificity and Its Applications in Retrieval. *Journal of Documentation* 28, 11-21.

Kamp, H. A. (1981). A Theory of Truth and Semantic Representation. *Formal Methods in the Study of Language,* Part 1, 277-322.

Kantrowitz, M., Mohit, B. & Mittal, V. (2000). Stemming and Its Effects on TFIDF Ranking. *Proceedings of ACM-SIGIR 2000*, Athens, Greece, pp. 357-359.

Klemettinen, M., Mannila, H., Ronkainen, P., Toivonen, H. & Verkamo, A. (1999). Finding Interesting Rules from Large Sets of Discovered Association Rules. *Proceedings of the 3rd International Conference on Information and Knowledge Management*, pp. 401-407.

Kodratoff, Y. (1999). Knowledge Discovery in Texts: A Definition and Applications. *Proceedings of the 11th International Symposium on Foundations of Intelligent Systems*, pp. 16-29.

Levenshtein, V. I. (1966). Binary Codes Capable of Correcting Insertions and Reversals. *Soviet Physics Doklady*, 10, 707-710.

Liu, B. & Hsu, W. (1996). Post-Analysis of Learned Rules. *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, Portland, Oregon, pp. 828-834.

Liu, B., Hsu, W., Mun, L. F. & Lee, H. Y. (1999). Finding Interesting Patterns Using User Expectation. *IEEE Transactions on Knowledge and Data Engineering,* 11, pp. 817-832.

Liu, B., Hsu, W., Chen, S. & Ma, Y. (2000). Analyzing the Subjective Interestingness of Association Rules. *IEEE Intelligent Systems* 15(5), 47-55.

Liu, B., Hsu, W. & Ma, Y. (2001). Identifying Non-Actionable Association Rules. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, San Francisco, CA, pp. 329-334.

Miller, R. C. & Bharat, K. (1998). SPHINX: A Framework for Creating Personal Site-Specific Web Crawlers. *Proceedings of WWW7*, Brisbane Australia, pp. 119-130.

Nahm, U. Y. & Mooney, R. J. (2000). Using Information Extraction to Aid the Discovery of Prediction Rules from Text. *Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining Workshop on Text Mining*, Boston, MA, pp. 51-58.

Nahm, U. Y. & Mooney, R. J. (2000). A Mutually Beneficial Integration of Data Mining and Information Extraction. *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, Austin, TX, pp. 627-632.

Nahm, U. Y. & Mooney, R. J. (2002). Text Mining with Information Extraction. *Proceedings of the AAAI 2002 Spring Symposium on Mining Answers from Texts and Knowledge Bases*, Stanford, CA, pp. 60-67.

Nahm, U. Y. & Mooney, R. J. (2002). Mining Soft-Matching Association Rules. *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, McLean, VA, pp. 681-683.

Ngai, G. & Yarowsky, D. (2000). Rule Writing or Annotation: Cost-efficient Resource Usage for Base Noun Phrase Chunking. *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, Hong Kong, pp. 117-125.

Padmanabhan, B. & Tuzhilin, A. (1998). A Belief-Driven Method for Discovering Unexpected Patterns. *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, New York, NY, pp. 94-100.

Padmanabhan, B. & Tuzhilin, A. (1999). Unexpectedness as a Measure of Interestingness in Knowledge Discovery. *Decision Support Systems*, 27, 303-318.

Piatesky-Shapiro, G. & Matheus, C. (1994). The Interestingness of Deviations. *Proceedings of the KDD-94 Conference*, pp. 25-36.

Pierre, J. M. (2002). Mining Knowledge from Text Collections Using Automatically Generated Metadata. *Proceedings of the Fourth International Conference on Practical Aspects of Knowledge Management,* pp. 537-548.

Salton, G. & McGill, M. J. (1983). *Introduction to Modern Information Retrieval.* McGraw-Hill.

Salton, G. & Buckley, C. (1988). Term Weighting Approaches in Automatic Text Retrieval. *Information Processing and Management*, 24(5), 513-523.

Sanderson, M. & Croft, B. (1999). Deriving Concept Hierarchies from Text. *Proceedings of the 22nd annual international ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 206-213.

Sang, E. k. F. T. K., Daelemans, W., H. Dejean, Koeling, R., Krymolowski, Y., Punyakanok, V. et al. (2000). Applying System Combination to Base Noun Phrase Identification. *Proceedings of COLING 2000*, pp. 857-863.

Shen, W., Ong, K., Mitbander, B. & Zaniolo, C. (1996). Metaqueries for Data Mining. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy (Eds.), *Advances in Knowledge Discovery and Data Mining* (375-398). AAAI/MIT Press.

Silberschatz, A. & Tuzhilin, A. (1995). On Subjective Measures of Interestingness in Knowledge Discovery. *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, pp. 275-281.

Silberschatz, A. & Tuzhilin, A. (1996). What Makes Patterns Interesting in Knowledge Discovery Systems. *IEEE Transactions on Knowledge and Data Engineering*, 8(6), 970-974.

Snow, C. E. & Ferguson, C. A. (1997). *Talking to Children: Language Input and Acquisition*. Cambridge: Cambridge University Press.

Tan, P., Kumar, V. & Srivastava, J. (2002). Selecting the Right Interestingness Measure for Association Patterns. *Proceedings of KDD'02*, pp. 32-41.

Tan, P., Kumar, V. & Srivastava, J. (2004). Selecting the Right Objective Measure for Association Analysis. *Information Systems*, 29(4), 293-313.

Turney, P. D. (2000). Learning Algorithm for Keyphrase Extraction. *Information Retrieval*, 2(4), 303-336.

Wu, Y. B. (2001). *Automatic Concept Organization: Organizing Concepts from Text through Probability of Co-occurrence Analysis (POCA)*. PhD thesis.

Wu, Y. B., Li, Q., Bot, R. S. & Chen, X. (2004). KIP: Keyphrase Identification Program with Learning Functions. *Proceedings of ITCC 2004*, Las Vegas.

Wu, Y. B., Bot, R. S., Chen, X. & Li, Q. (2005). Improving Concept Hierarchy Development for Web Returned Documents Using Automatic Classification. *Proceedings of ICOMP'05*, Las Vegas, USA.