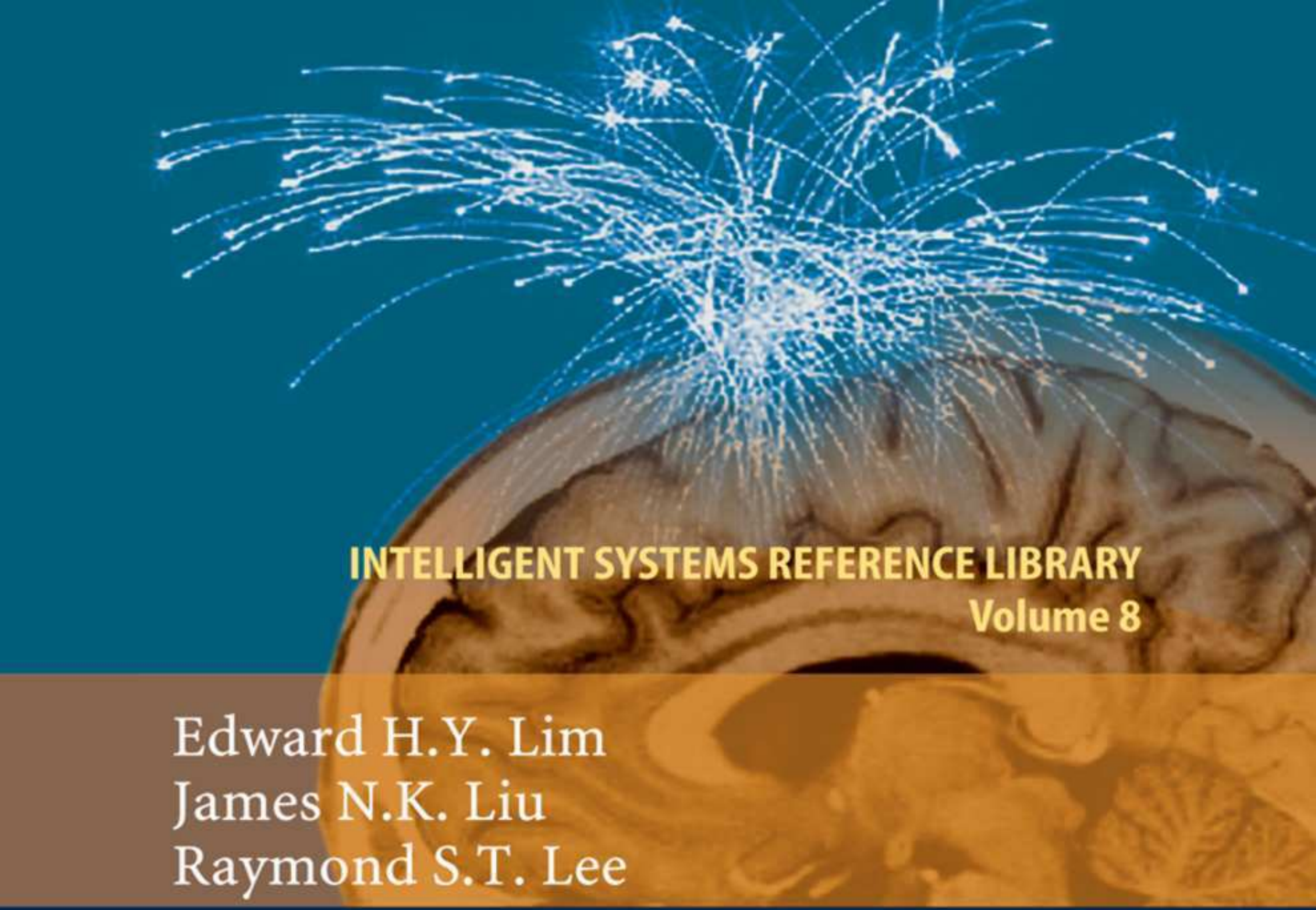


INTELLIGENT SYSTEMS REFERENCE LIBRARY
Volume 8

Edward H.Y. Lim
James N.K. Liu
Raymond S.T. Lee

Knowledge Seeker – Ontology Modelling for Information Search and Management

A Compendium

Springer

Edward H.Y. Lim, James N.K. Liu, and Raymond S.T. Lee

Knowledge Seeker – Ontology Modelling for Information Search and Management

Intelligent Systems Reference Library, Volume 8

Editors-in-Chief

Further volumes of this series can be found on our homepage: springer.com

Vol. 1. Christine L. Mumford and Lakhmi C. Jain (Eds.)
Computational Intelligence: Collaboration, Fusion and Emergence, 2009
ISBN 978-3-642-01798-8

Vol. 2. Yuehui Chen and Ajith Abraham
Tree-Structure Based Hybrid Computational Intelligence, 2009
ISBN 978-3-642-04738-1

Vol. 3. Anthony Finn and Steve Scheding
Developments and Challenges for Autonomous Unmanned Vehicles, 2010
ISBN 978-3-642-10703-0

Vol. 4. Lakhmi C. Jain and Chee Peng Lim (Eds.)
Handbook on Decision Making: Techniques and Applications, 2010
ISBN 978-3-642-13638-2

Vol. 5. George A. Anastassiou
Intelligent Mathematics: Computational Analysis, 2010
ISBN 978-3-642-17097-3

Vol. 6. Ludmila Dymowa
Soft Computing in Economics and Finance, 2011
ISBN 978-3-642-17718-7

Vol. 7. Gerasimos G. Rigatos
Modelling and Control for Intelligent Industrial Systems, 2011
ISBN 978-3-642-17874-0

Vol. 8. Edward H.Y. Lim, James N.K. Liu, and Raymond S.T. Lee
Knowledge Seeker – Ontology Modelling for Information Search and Management, 2011
ISBN 978-3-642-17915-0

Edward H.Y. Lim, James N.K. Liu, and
Raymond S.T. Lee

Knowledge Seeker – Ontology Modelling for Information Search and Management

A Compendium

Springer

Edward H.Y. Lim
Department of Computing
The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong
E-mail: edward@iatopia.com

Raymond S.T. Lee
IATOPIA Research Lab
Suite 1515, Star House, 3 Salisbury Road
Tsim Sha Tsui, Kowloon, Hong Kong
E-mail: raymond@iatopia.com

James N.K. Liu
Department of Computing
The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong
E-mail: csnkliu@inet.polyu.edu.hk

ISBN 978-3-642-17915-0 e-ISBN 978-3-642-17916-7

DOI 10.1007/978-3-642-17916-7

Intelligent Systems Reference Library ISSN 1868-4394

Typeset & Cover Design: Scientific Publishing Services Pvt. Ltd., Chennai, India.

Printed on acid-free paper

9 8 7 6 5 4 3 2 1

springer.com

Preface

Ontology is being a fundamental form of knowledge representation about the real world. In the computer science perspective, ontology defines a set of representational primitives with which to model a domain of knowledge or discourse (Gruber 2008). A well constructed ontology can help developing knowledge-based information search and management system, such as search engine, automatic text classification system, content management system, etc, in a more effective way. Most of these existing systems are ineffective in terms of its low accuracy in searching and managing information (especially text data), because of lacking knowledge as the core components used in the systems. Ontology is therefore becoming a very important research area for developing those knowledge-based information systems, as ontology is a recognized form of representing a particular knowledge domain.

The most challenge of ontology research is how to create and maintain ontology. Ontology engineering is a kind of this ontology research for developing theories, methods, and software tools that help creating and maintaining ontology. The ontology engineering methods developed in the past mainly consist of manual ontology creation and automatic ontology learning methods. Although many ontology engineering tools have been developed over the last decade, most of them involve manual creating and maintaining ontology which is a time consuming and inefficient task as every process requires deep analysis by domain experts. There is also a problem that domain experts may create ontology by different and subjective view, so that the ontology knowledge is not exact and may not be relevant to all knowledge domains. Therefore, automatic ontology learning is a more practical method in ontology engineering. This ontology learning method tried to develop an automated process to extract knowledge from some computer data and present as a specific form of ontology, with the least or minimum involvement of human work.

Ontology learning from text is the most useful method in formalizing ontology, as text data is a rich and direct source of human knowledge. However, analyzing textual data by computer is a difficult task, as it requires some natural language processing and semantic analysis. Many methodologies on ontology learning from text have been widely developed in recent years (Maedche 2002, Buitelaar and Cimiano 2008). Most of them use artificial intelligent approaches such as machine learning or statistical analysis to develop the methodologies, and try to extract ontology from text learning automatically.

A lot of researches on ontology learning in text have been carried out in recent years. However, most of them applied only on English text, as text is a language dependent data, algorithms applied on English text were found not working well in Chinese text. Therefore, the research and some related experiments on ontology learning in Chinese text are focused in this project, as we aim to develop an efficient ontology learning method which can be applied to Chinese text data. Information search and management systems that contain mainly Chinese text data hence can be enhanced by the ontology, because many existing ontology are developed in English which cannot be applied to Chinese based information system. In this research project, we aim to develop a comprehensive system framework of ontology learning in Chinese text which can be applied to Chinese based information search and management systems.

The overall objective of this research is to develop an ontology based system framework, called KnowledgeSeeker, which can be used to develop various ontology based information systems. These information systems mainly include Chinese based text information retrieval system, such as search engine and content management system. The final goal of the KnowledgeSeeker system framework is that it can improve the traditional information system with higher efficiency. In particular, it can increase the accuracy of a text classification system, and also enhance the search intelligence in a search engine. This can be done by enhancing the system with machine processeable knowledge (ontology). It has been mentioned that ontology is a useful component in developing knowledge based intelligent information systems, but the problem is that lots of research work are still required to find out the method of creating and maintaining a relevant ontology for used in the information system. Therefore we raise the following research questions to define the scope of this research work:

- What format of the ontology can be modeled in computer system?
- What learning method is used to automatically create the ontology data?
- How to generate the ontology for machine and also be visualized for human use?
- What operations can be done with the ontology and how it operates?
- What applications can be developed by using the ontology and how is the performance?

This book is organized in three parts containing twelve chapters:

Part I (Chapters 1-4): Introduction
Part II (Chapters 5-8): KnowledgeSeeker - An Ontology Modeling and Learning Framework
Part III (Chapter 9s-12): KnowledgeSeeker Applications

The book is outlined as follows:

Chapter 1 presents briefly the philosophical question about knowledge and ontology, and also the perspective of ontology in information system.

Chapter 2 presents the ontology engineering approaches in the recent researches, including the fundamental concepts about ontology and ontology learning.

Chapter 3 presents the traditional text information retrieval system and related models and algorithms.

Chapter 4 presents the research about web data semantics and some semantic modeling technologies for developing intelligent systems.

Chapter 5 presents an overview of the system framework called Knowledge-Seeker. It also presents the proposed graphical based model of ontology called Ontology Graph.

Chapter 6 presents the methodology of automatic ontology learning in Chinese text. Relevant examples and experiments are presented to illustrate the proposed methodology.

Chapter 7 presents the Ontology Graph generation process.

Chapter 8 presents different kinds of Ontology Graph based operation and operating methods.

Chapter 9 presents an application, a text classification application with experiment, which adopts the technology of KnowledgeSeeker for classification. It provides experimental result and performance analysis.

Chapter 10 presents a commercial application which adopts the techniques of KnowledgeSeeker system framework called IATOPIA iCMS KnowledgeSeeker, which is ontology based digital assets management system for managing multimedia files.

Chapter 11 presents another commercial application called IATOIPA News Channel (IAToNews), which is an ontology-based news system provided in web environment.

Chapter 12 presents a collaborative content and user-based web ontology learning system, which enhances the ontology learning method by user-based knowledge.

Acknowledgements

Upon the completion of this book, we would like to take the opportunity to express our sincere gratitude to the following parties or persons. We are grateful for the partial support of GRF grant 5237/08E and CRG grant G-U756 of Department of Computing, The Hong Kong Polytechnic University. We also thank IATOPIA Research Lab for providing the facilities and convenience for system testing. Lastly, we thank Prof. Lakhmi Jain of the KES Centre, University of South Australia who encouraged us to carry out the book.

Edward H.Y. Lim[1,2], James N.K. Liu[1], and Raymond S.T. Lee[2]

[1] Department of Computing
The Hong Kong Polytechnic University
[2] IATOPIA Research Lab

Contents

Part I: Introduction

1 Computational Knowledge and Ontology 3

1.1 What Is Knowledge?..3

1.2 Computational Knowledge in Information Systems 3

1.2.1 Knowledge Engineering 4

1.2.2 Knowledge Representation 4

1.3 What Is Ontology?...6

1.4 Ontology Modeling in Computer Systems..7

1.4.1 Computational Ontology Representation...7

1.4.2 Top-Level Ontologies 8

1.4.3 Lexical Ontologies...9

1.4.4 Domain Ontologies 11

2 Ontology Engineering...13

2.1 Introduction 13

2.2 Ontology Fundamentals..14

2.2.1 Ontological Structure 14

2.2.2 Ontological Taxonomy 15

2.2.3 Ontological Relations 16

2.2.4 Lexical Relations 18

2.3 Ontology Engineering Tools...19

2.3.1 Protégé ontology editor...20

2.4 Ontology Learning from Text..21

2.4.1 Learning Terminology 23

2.4.2 Learning Taxonomy..25

3 Text Information Retrieval...27

3.1 Information Retrieval Model 27

3.1.1 Term Weighting Model...27

3.1.2 Text Classification 28

3.1.3 Text Clustering 31

3.2 Feature Selection and Reduction 32
3.3 Search Engine Modeling 34
3.4 Evaluation Methods 35
3.4.1 Performance measurement 36

4 Web Data Semantics 37
4.1 Semantic Web 37
4.1.1 W3C Semantic Web 39
4.2 Semantic Modeling 40
4.2.1 Taxonomy 41
4.2.2 Thesaurus 41
4.2.3 Topic Maps 42
4.2.4 Ontology 43
4.2.5 Ontology Languages for the Semantic Web 44
4.3 Semantic Annotation and Information Retrieval 46

Part II: KnowledgeSeeker: An Ontology Modeling and Learning Framework

5 Ontology Modeling Framework 49
5.1 KnowledgeSeeker – The System Overviews 49
5.2 Background of Signs System and Ontology 51
5.2.1 The Semiotics 51
5.2.2 The Sign System for Concept Formation 54
5.2.3 The Meaning of Words 56
5.2.4 The Semantics of Relations 57
5.3 Ontology Graph Overviews 59
5.3.1 Nodes in Ontology Graph 60
5.3.2 Term Nodes in Ontology Graph 62
5.3.3 Words Function 64
5.3.3.1 The Function of Language 64
5.3.3.2 Understanding and Feeling 65
5.3.3.3 Meaning and Information 67
5.4 The Implementation of Ontology Graph 68
5.4.1 The Conceptual Structure of Ontology Graph 68
5.4.1.1 Conceptual Units in Ontology Graph 69
5.4.2 The Class Diagram of Ontology Graph 69

6 Ontology Learning in Chinese Text 71
6.1 The Ontology Learning Method 71
6.1.1. Term Extraction 72
6.1.2. Term-to-Class Relationship Mapping 73
6.1.2.1 Term-to-Class Independency Measurement by χ^2 73
6.1.2.2 Term-to-Class Positive and Negative Dependency Measurement by R 78

6.1.3. Term-to-Term Relationship Mapping 81
6.1.3.1 Term-to-Term Independency Measurement by χ^2 83
6.1.4. Concept Clustering 91
6.1.5 Sample Result of Domain Ontology Graph Generation (10 Domains) 96

7 Ontology Graph Generation Process 99
7.1 Ontology in Information Systems 99
7.2 Ontology Graph Generation Process in KnowledgeSeeker 100
7.2.1 Definition of Ontology Graph Structure 101
7.2.2 Domain Ontology Graph Generation Process..............................103
7.2.3 Document Ontology Graph Generation 105
7.3 Automatic Generation of Domain Ontology Graph...............................107
7.3.1 Experimental Setup...108
7.3.2 Experimental Results 110

8 Ontology Graph Operations 121
8.1 Ontology Graph Matching and Querying Process 121
8.2 Introduction to Ontology Matching and Mapping 121
8.2.1 Ontology Graph Matching Methods 122
8.2.1.1 Semantic Mapping Function 122
8.2.1.2 Similarity Measurement Function..................................123
8.3 Matching Different Components with Ontology Graph 124
8.3.1 Matching Terms to Domain Ontology Graph 125
8.3.2 Matching Text Document to Domain Ontology Graph 128
8.3.3 Ontology Graph Based Similarity Measurement 132
8.3.4 Matching Two Document Ontology Graphs................................134
8.4 Overviews of Ontology Graph Based Querying 137
8.4.1 Ontology Graph Querying Methods 137
8.5 Operations in Ontology Graph Based Querying 137
8.5.1 Querying Document with Document Ontology Graph 138

Part III: KnowledgeSeeker Applications

9 Ontology Graph Based Approach for Automatic Chinese Text Classification 145
9.1 Introduction 145
9.2 Methodologies 147
9.2.1 Ontology Graphs Reviews 147
9.2.2 Classification Algorithm..148
9.3 Experiments 149
9.3.1 Experiment Description 150
9.3.1.1 Evaluate the Performance of Ontology-Graph Based Approach (Experiment 1)..150

9.3.1.2 Evaluate the Optimal Size of Domain Ontology Graph for Best Classification Result (Experiment 2)..............151

9.3.1.3 Evaluate the Effects for Setting Different Thresholds of weight of Domain Ontology Graph (Experiment 3) 151

9.3.2 Evaluation Method..151

9.3.3 Performance on Ontology Graph Based Text Classification 152

9.3.4 Experimental Results 153

9.3.4.1 Performance on Ontology Graph Based Text Classification (Experiment 1) 153

9.3.4.2 Performance on Using Different Size of Terms (Dimensionality) for Text Classification (Experiment 2) 156

9.3.4.3 Result of Using Different Thresholds of Weight of Domain Ontology Graphs (Experiment 3).....................161

9.3.4.4 Combining the Results and Optimizing the Parameters for the Text Classification Experiments 163

10 IATOPIA iCMS KnowledgeSeeker – An Integrated Content Management System and Digital Asset Management System (DAMS)..165

10.1 IATOPIA iCMS KnowledgeSeeker..165

10.1.1 System Features 165

10.1.2 System Model and Architecture...166

10.1.3 Ontology System...168

10.1.4 Search Indexing System...168

10.1.5 IATOPIA Search Engine 169

10.2 IATOPIA Digital Asset Management System (DAMS) 170

10.2.1 DAMS System Architecture Overview...................................170

10.2.2 IATOIPA iCMS Databank Cluster in DAMS.........................170

10.2.3 Ontology System in DAMS 171

10.2.4 DAMS and Web Channel Interface Examples........................172

11 IATOPIA News Channel (IAToNews) – An Intelligent Ontological Agent-Based Web News Retrieval and Search System...........................175

11.1 Introduction..175

11.2 IAToNews System Architecture Overview 175

11.3 Ontology System in IAToNews..176

11.3.1 Article Ontology 177

11.3.2 Topic Ontology..178

11.3.3 Ontology Based Content Indexing..178

11.4 IAToNews Web Interface Examples 179

12 Collaborative Content and User-Based Web Ontology Learning System 181
12.1 Introduction 181
12.2 Background 182
12.2.1 Problem of Building a Generalized Ontology to Satisfy Daily Life 182
12.2.2 Semantic Web 182
12.2.3 Web Channels 182
12.2.4 BuBo (Feedback, Personalization with User Ontology) 183
12.3 Methodology 183
12.3.1 Overview of System Architecture 183
12.3.2 Content-Based Ontology Learning Process 184
12.3.3 User-Based Ontology Personalization Process 187
12.4 Implementation 192
12.4.1 Architecture of the Collaborative System Implementation 192
12.4.2 Structure of the Specified Domains Ontology with Generalized OG 193
12.4.3 Ontology-Based Search Engine Within BuBo 194
12.5 Conclusions 194

References 195

Appendix 211

List of Figures

1.1 SUMO top level 8
1.2 OpenCyc upper-level ontology hierarchy (OpenCyc 2003) 9
1.3 Graphical expression of Chinese lexical concept in HowNet 10
1.4 Basic data - Taxonomy in HowNet 11
1.5 Ontology sample of entertainment domain 12

2.1 Subject-relation-object representation 15
2.2 Taxonomical relations and non-taxonomical relations in ontology 15
2.3 Taxonomy and partonomy relations in ontology 16
2.4 Taxonomy and partonomy relations in ontology 16
2.5 Ontological relations by universals and particulars example 17
2.6 Ontological relations by universals and particulars example 18
2.7 The Protégé-Frames editor for OKBC ontology modeling 20
2.8 The Protégé-OWL editor for OWL ontology modeling 21
2.9 Degree of knowledge formalization in seven levels 22

3.1 8 x 7 term-by-document matrix for 7 documents and 8 terms 28
3.2 kNN classification approach 30
3.3 Inputs, output and function in a neuron 30
3.4 3-layer structure of neural network classification 31
3.5 Search engine process and its components 34

4.1 Classification model and its sub-processes 38
4.2 Semantic Web stack 40
4.3 The semantic modeling techniques in levels 40
4.4 Example of a financial site map organized in a taxonomy hierarchy 41
4.5 The semantic modeling techniques in levels 42
4.6 XTM example 43
4.7 Language for ontology modeling 44
4.8 RDF subject-predicate-object triple 44
4.9 RDF triple relations 44
4.10 RDF example 45
4.11 OWL example 45
5.1 Four modules in KnowledgeSeeker system framework 49
5.2 Knowledge components of ontology learning from text 50
5.3 A sign represented by Icons 52

5.4 Saussure's sign model 53
5.5 Signification between concept and word 53
5.6 Signification between concept and word 53
5.7 Relations between signs 54
5.8 The Meaning Triangle (Ogden and Richards 1923) 54
5.9 The meaning triangle of the symbol "Dog" 55
5.10 Signs and relations 55
5.11 Ontological triples 56
5.12 Word network by word triples 57
5.13 Sentence triple for word network 57
5.14 Conceptual units as a network model 60
5.15 A formation of meaning with terms - example 61
5.16 A formation of meaning with symbols - example 61
5.17 Example of meaning formation 64
5.18 Jakobson's nodel of the function of language 64
5.19 The functions of language and knowledge 65
5.20 The functions of language and knowledge 66
5.21 Relation types expressing different nature of knowledge 67
5.22 Conceptual structure of Ontology Graph (OG) in KnowledgeSeeker 68
5.23 The class relationship of Ontology Graph 70

6.1 Bottom-up approach of the Ontology Graph learning process 72
6.2 Term mapping to classes 73
6.3 Term-to-class Relationship mapping steps 74
6.4 8 x 10 term-by-document matrix for 10 documents and 8 terms 75
6.5 Term mapping to classes from the example 76
6.6 Updated term-to-class relationship mapping steps 79
6.7 Terms mapping to each other 82
6.8 Create document links to each term in the term-group 86
6.9 Term-to-term relationship mapping steps 87
6.10 Ontology Graph created for 4 terms in class c_1 *(k=4)* 91
6.11 Terms-to-concept clustering steps 92
6.12 Final Ontology Graph created for class c_1 92
6.13 The distribution of terms in their POS 96
6.14 *DOG* (文藝 Arts and Entertainments) 97
6.15 *DOG* (政治 Politics) 97
6.16 *DOG* (政治 Traffic) 97
6.17 *DOG* (教育 Education) 97
6.18 *DOG* (環境 Environment) 97
6.19 *DOG* (經濟 Economics) 97
6.20 *DOG* (經濟 Military) 98
6.21 *DOG* (醫療 Health and Medical) 98
6.22 *DOG* (電腦 Computer and Information Technology) 98
6.23 *DOG* (體育 Sports) 98

7.1 Three types of Ontology according to their level of abstraction................100
7.2 Types of ontology in different levels..100
7.3 Domain Ontology Graph generation process..103
7.4 A graphical representation of the *DOG* of "*entertainment*" 105
7.5 Document Ontology Graph generation process...106
7.6 The steps of automatic domain ontology graph generation 107
7.7 The interface of yED Graph Editor (yED 2009)..108
7.8 Sample of a generated GraphML document of *DOG* 109
7.9 Number of nodes generated for different threshold values........................110
7.10 The rates of nodes generated for different threshold values 110
7.11 Number of edges generated for different threshold setting........................112
7.12 The rate of edges generated for different threshold setting........................112
7.13 $\theta_u = 20$, $\theta_v = 200$..........114
7.14 $\theta_u = 20$, $\theta_v = 150$..........114
7.15 $\theta_u = 20$, $\theta_v = 100$..........114
7.16 $\theta_u = 20$, $\theta_v = 90$..........114
7.17 $\theta_u = 20$, $\theta_v = 80$..........115
7.18 $\theta_u = 20$, $\theta_v = 70$..........115
7.19 $\theta_u = 20$, $\theta_v = 60$..........116
7.20 $\theta_u = 20$, $\theta_v = 50$..........116
7.21 $\theta_u = 20$, $\theta_v = 40$..........117
7.22 $\theta_u = 20$, $\theta_v = 30$..........117
7.23 $\theta_u = 20$, $\theta_v = 20$..........117
7.24 $\theta_u = 20$, $\theta_v = 10$..........117
7.25 $\theta_u = 20$, $\theta_v = 0$..........117
7.26 $\theta_u = 120$, $\theta_v = 200$..........118
7.27 $\theta_u = 120$, $\theta_v = 160$..........118
7.28 $\theta_u = 120$, $\theta_v = 120$..........118
7.29 $\theta_u = 120$, $\theta_v = 100$..........118
7.30 $\theta_u = 120$, $\theta_v = 80$..........118
7.31 $\theta_u = 120$, $\theta_v = 60$..........118
7.32 $\theta_u = 120$, $\theta_v = 40$..........119
7.33 $\theta_u = 120$, $\theta_v = 20$..........119

8.1 Ontology matching overviews 122
8.2 Terms mapping example..122
8.3 Components of Ontology Graph mapping process 124
8.4 Components of concept formation process...125
8.5 Process of concept formation...125
8.6 Ontology Graph OG_A..126
8.7 *OG* of Concept c_A..127
8.8 *OG* of Concept c_B..127
8.9 *OG* of Concept c_C 127
8.10 *OG* of Concept c_D 128
8.11 *OG* of Concept c_E..128

8.12 Mapping overlapping terms in document and Domain Ontology Graph 128
8.13 Components of Document Ontology Graph generation process 129
8.14 Process of Document Ontology Graph (*DocOG*) extraction 129
8.15 *DocOG* for $d_1 - OG_{d_1}$ 132
8.16 *DocOG* for $d_2 - OG_{d_2}$ 132
8.17 Components of Document and Domain Ontology Graph comparison 133
8.18 Process of Document and Domain Ontology Graph comparison 133
8.19 Components of comparison between two documents 135
8.20 Process of documents comparisons 135
8.21 OG of mapping result 136
8.22 Ontology querying overviews 137
8.23 Components of Ontology Graph querying process 138
8.24 Components of document ontology graph formation process 139
8.25 Process of Document Ontology Graph extraction 139
8.26 *DocOG* for $d_1 - OG_{d_1}$ 140
8.27 *DocOG* for $d_2 - OG_{d_2}$ 141

9.1 Conceptual structure of Ontology Graph 147
9.2 Comparison of document and domain ontology graph for similarity measurement 148
9.3 Scoring terms inside a document 149
9.4 Result of precision for different approaches 155
9.5 Result of recall for different approaches 155
9.6 Result of F-measure for different approaches 156
9.7 Result of precision and recall for *tf-idf* approach 157
9.8 Result of precision, recall and f-measure for *tf-idf* approach 158
9.9 Result of precision and recall for term-dependency approach 159
9.10 Result of precision, recall and f-measure for term-dependency approach 159
9.11 Result of precision and recall for ontology-graph approach 160
9.12 Result of precision, recall and f-measure for ontology-graph approach 161
9.13 Precision and recall for different threshold values used 162
9.14 Result of classification in precision, recall and f-measure 163
9.15 Result comparison of precision and recall for the three approaches 163
9.16 Result comparison of f-measure for the three approaches 164

10.1 The system architecture of iCMS KnowledgeSeeker 166
10.2 Process flow of IATOPIA iCMS KnowledgeSeeker 167
10.3 DAMS architecture overview 170
10.4 IATOPIA iCMS databank cluster for DAMS 171
10.5 DAMS interface for movie domain 172
10.6 Web channel linked to DAMS content for movie domain 172
10.7 Example of Chinese Opera ontology tree maintained by domain experts 173

11.1 IAToNews system architecture and information flow 176
11.2 Ontology system in IAToNews 177
11.3 Article ontology in IAToNews ontology system 178
11.4 Main page displaying classified news in IAToNews.................................179
11.5 Display news content and related info in IAToNews 180

12.1 Ontology agent application..184
12.2 Four content-based ontology learning processes 184
12.3 Ontology Graph 185
12.4 Ontology graph containing *k* number of Concepts 187
12.5 Structure of the user preferences XML...188
12.6 The top level Ontology Graph 189
12.7 The example structure of Ontology Graph 189
12.8 SG interception with the POG and form new POG 190
12.9 The process of customization and how the user uses the BuBo to interact with the User's Personalization Ontology Search Agent 191
12.10 The ontology learning server and web channel server..............................192
12.11 BuBo Structure and responsibilities...193
12.12 The example of the topic-base generalized OG 193

List of Tables

1.1 Knowledge representation formalisms (Guarino 1995) 5
1.2 Aristotle's categories 6
1.3 Kant's categories 6

2.1 Transitive relations characteristic 18

3.1 Content of terms and documents 28
3.2 Term-to-category contingency table 33
3.3 The table of retrieval result 35

4.1 Examples of semantic relations in thesaurus 42
4.2 Components of an ontology 43

5.1 Coherence Relations (Louwerse 2001, Mancini and Shum 2006) 59
5.2 Summary of POS in IG 63
5.3 Types of relations - example 63
5.4 Factors of Communication and Functions of Language 65
5.5 Referential and Emotive function of Verb and Adjective 66
5.6 Factors of Communication and Functions of Language 67
5.7 Class Relations of Ontology Graph Implementation 69

6.1 2 x 2 term-to-class contingency table of term t and class c 73
6.2 Content of term and documents distribution 75
6.3 Term-to-class table - 10 documents in 5 classes with 8 candidate terms 75
6.4 χ^2 mapping of 5 classes and 8 candidate terms 76
6.5 2x2 term-to-class contingency table of term t_6 and class c_1 77
6.6 2x2 term-to-class contingency table of term t_6 and class c_3 77
6.7 2x2 term-to-class contingency table of term t_6 and class c_1 77
6.8 2x2 term-to-class contingency table of term t_6 and class c_3 77
6.9 $R_{t,c}$ mapping of 5 class and 8 candidate terms 79
6.10 Dependency values of term t_1 80
6.11 Dependency values of term t_2 80
6.12 Dependency values of term t_3 80

6.13 Dependency values of term t_4 80
6.14 Dependency values of term t_5 80
6.15 Dependency values of term t_6 80
6.16 Dependency values of term t_7 80
6.17 Dependency values of term t_8 80
6.18 2x2 term-to-term contingency table of term t_a and term t_b 83
6.19 Term dependency $n\chi^{2}_{t_a,t_b}$ of class c_i 84
6.20 Term dependency R_{w_a,w_b} of class c_i 84
6.21 Term-groups created for class c_1 to c_5 85
6.22 Content of terms and documents distribution of the document
corpus D_2 86
6.23 Document link from D_2 to C_{c1} 88
6.24 Term-to-term table (10 documents for the term-group of class c_1) 88
6.25 2x2 term-to-term contingency table of term t_1 and term t_1 88
6.26 2x2 term-to-term contingency table of term t_1 and term t_2 89
6.27 2x2 term-to-term contingency table of term t_1 and term t_3 89
6.28 2x2 term-to-term contingency table of term t_6 and term t_4 89
6.29 χ^{2} term-to-term mapping of 4 terms in class c_1 90
6.30 Final Result - terms dependency $n\chi^{2}_{t_a,t_b}$ of class c_1 90
6.31 Final Result - terms dependency R_{t_a,t_b} of class c_1 90
6.32 Class label (Chinese & English) in document corpus D_1 93
6.33 Document distribution of the ten classes (D_1) 93
6.34 Document distribution for learning and testing 94
6.35 Documents distribution of corpus D_2 94
6.36 Statistics of term extraction results 95
6.37 Results of term-to-class relationship mapping 95
6.38 Ratio of positive to negative terms in each class 96

7.1 Term list of a sample DOG of domain “entertainment” 104
7.2 Details of the rates of nodes generated for different thresholds values 111
7.3 Top 20 terms of domain “文藝” in ontology learning 113
7.4 DOC generation result of the domain “文藝 (Arts and Entertainments)” 113

8.1 Table of the terms and relations in OG_A 126
8.2 Mapping of concept c_A 127
8.3 Mapping of concept c_B 127
8.4 Mapping of concept c_C 127
8.5 Mapping of concept c_D 128
8.6 Mapping of concept c_E 128
8.7 $c_{A},_{d_1}$ 130
8.8 $c_{B},_{d_1}$ 130
8.9 $c_{D},_{d_1}$ 131

8.10 $c_{D}, _{d_2}$ 131
8.11 $c_{E}, _{d_2}$ 131
8.12 Terms and relations in OG_{d_1} 132
8.13 Terms and relations in OG_{d_2} 132
8.14 Terms and relations in OG_{d_1} 134
8.15 Terms and relations in OG_{d_2} 134
8.16 Mapping result..136
8.17 Terms and relations in OG_{d_1} 140
8.18 Terms and relations in OG_{d_2} 141
8.19 Result of q_1-to-OG_{d_1} 141
8.20 Result of q_1-to-OG_{d_2} 141
8.21 Result of q_2-to-OG_{d_1} 141
8.22 Result of q_2-to-OG_{d_2} 141
8.23 Result of q_3-to-OG_{d_1} 141
8.24 Result of q_3-to-OG_{d_2} 141

9.1 The table of retrieval result..151
9.2 Class label (Chinese & English) of training document set 152
9.3 Document count for training and testing document sets..153
9.4 Details of classification result of each class..154
9.5 Classification result for *tf-idf* approach..157
9.6 Classification result for term-dependency approach..158
9.7 Classification result for ontology-graph approach..160
9.8 Experimental result of using different threshold values 162
9.9 Summary of the optimized performance..164

11.1 Article data in article ontology 177
11.2 Semantic data in article ontology..177
11.3 Topics in IATOPIA KnowledgeSeeker news channel..178
11.4 Types of ontology based content index of news article..179

A.1 Top 30 ranked terms selected in Domain Ontology Graph (文藝)...........213
A.2 Top 30 ranked terms selected in Domain Ontology Graph (政治)...........214
A.3 Top 30 ranked terms selected in Domain Ontology Graph (交通)...........215
A.4 Top 30 ranked terms selected in Domain Ontology Graph (教育)...........216
A.5 Top 30 ranked terms selected in Domain Ontology Graph (環境)...........217
A.6 Top 30 ranked terms selected in Domain Ontology Graph (經濟)...........217
A.7 Top 30 ranked terms selected in Domain Ontology Graph (軍事)...........219
A.8 Top 30 ranked terms selected in Domain Ontology Graph (醫療)...........220
A.9 Top 30 ranked terms selected in Domain Ontology Graph (電腦)...........221
A.10 Top 30 ranked terms selected in Domain Ontology Graph (體育)...........222

A.11 Terms dependency in Domain Ontology Graph (文藝)............................223
A.12 Terms dependency in Domain Ontology Graph (政治)............................224
A.13 Terms dependency in Domain Ontology Graph (交通)............................225
A.14 Terms dependency in Domain Ontology Graph (教育)............................226
A.15 Terms dependency in Domain Ontology Graph (環境)............................227
A.16 Terms dependency in Domain Ontology Graph (經濟)............................228
A.17 Terms dependency in Domain Ontology Graph (軍事)............................229
A.18 Terms dependency in Domain Ontology Graph (醫療)............................230
A.19 Terms dependency in Domain Ontology Graph (電腦)............................231
A.20 Terms dependency in Domain Ontology Graph (體育)............................232

Part I
Introduction

Chapter 1
Computational Knowledge and Ontology

Abstract. The definition of knowledge is an augmentative topic in philosophy. We do not try to find out an explicit meaning of philosophical knowledge, but the most important thing is that we should know about what is knowledge in computer system, as called computational knowledge. In this chapter, we introduce some researches and definitions related to knowledge and computational knowledge. Ontology is a word used in both philosophy and computer system to describe the formalization of knowledge. We shall look into the definition of ontology in brief and also introduce its formalization methods in computer system.

1.1 What Is Knowledge?

“Knowledge” has been discussed by many philosophers since the Greek ancient times. It is not an easy task to find out a high abstraction of definition about “knowledge”. However, in very generally speaking, knowledge can be said as a meaningful resource that makes us know about the world. Theories of knowledge define what is about the world, how is it encoded, and in what way we reason about the world. Similar definition can be applied in computer and information system, unless we are defining those in the aim of computer processing instead of human understanding, and it is called computational knowledge.

1.2 Computational Knowledge in Information Systems

Computational knowledge in computer system has been represented as a hierarchy of data-information-knowledge in many knowledge management theories (Daft, 2004; Devenpart and Prusak 2000). Data refers to a string of bits, numbers or symbols that are only meaningful to a program. Data with meaning, such as computing words, texts, database records, etc, define information that is meaningful to human. Knowledge is the highest level of abstraction, which is encoded in some form inside information.

Creating computational knowledge is a study of artificial intelligent (AI) - an area of computer science focusing on making a computer to perform tasks with more intelligence (Genesereth and Nilsson 1987). Advanced information systems, such as information retrieval system, forecasting system, resource management system, online shopping system, personalization system, etc, always require computational knowledge to perform tasks with more intelligence. Traditional

E.H.Y. Lim, J.N.K. Liu, and R.S.T. Lee: Knowledge Seeker, ISRL 8, pp. 3 – 12.
springerlink.com

information systems are lacking of intelligence because they process data and information without analyzing the knowledge behind. To enable a computer understand and process knowledge, we need to discover and represent the knowledge from raw data to a computable form for processing. Intelligent information system with the ability to process knowledge is so called a knowledge-based system.

1.2.1 Knowledge Engineering

Knowledge engineering grew out rapidly with the increased desire of knowledge-based system in the past decade. Knowledge engineering is a process to find out a way or approach to extract useful knowledge from computer data. It requires processes of analyzing and discovering patterns of data and transforming them to a format that is understandable to either human or computer, or both. Over the years, knowledge engineering researches have been focusing on the development of theories, methods, and software tools which aid human to acquire knowledge in computer. They use scientific and mathematical approaches to discover the knowledge. The approaches can be simply defined as an input-process-output system: Input – the set of computer data such as texts and database records; process – the method for the transformation of input data to knowledge; output – the desired knowledge in a specific form of knowledge representation (such as ontology).

1.2.2 Knowledge Representation

A general view of knowledge representation can be summarized in five basic principles (Randall et al. 1993):

1. A knowledge representation is a surrogate - a substitute of a thing (a physical object, event and relationship) itself for reasoning about the world.
2. A knowledge representation is a set of ontological commitments - an ontology describing existences, categories, or classification systems about an application domain.
3. A knowledge representation is a fragmentary theory of intelligent reasoning - a theory of representation that supports reasoning about the things in an application domain. An explicit axioms or computational logic may be defined for intelligent reasoning.
4. A knowledge representation is a medium for efficient computation - other than the knowledge represented logically. It also must be encoded in some sort of format, language, which enables a computer to process it efficiently.
5. A knowledge representation is a medium of human expression – a knowledge representation that can be understood by human. It is used by knowledge engineers or domain experts to study and verify the knowledge.

In the area of information system, knowledge representation defines a computable form of knowledge in Computer. It applies the theories and techniques from other fields including (Sowa 1999): 1. Logic, it defines a formal structure and rules; 2. Ontology, it defines the kinds of existence in a domain of interest; and 3.

Computation, it supports and distinguishes knowledge representation from philosophical knowledge. A knowledge representation defines different types of knowledge, typically ontologies, facts, rules, and constraints (Chein and Mugnier 2008). Knowledge is represented independently of programming logic, which means it should be defined generic enough for use with different kinds of program and system. Therefore it requires a formalized and structuralized approach to develop a valid knowledge representation.

Knowledge representation language is used to express knowledge in information system. It can be classified according to the kinds of primitives used by user (Guarino 1995), as summarized in Table 1.1. They are also described in five different levels:

1. The logical level contains the basic primitives including predicates and functions. It is level of formalization allowing for a formal interpretation of the primitives.
2. The epistemological (Brachman 1979) level is a knowledge structure to fill the gap between logical levels, which are general and abstract primitives, and the conceptual level, which is a model of specific conceptual meaning.
3. The ontological level is an ontological commitment including ontological relations, associated to an explicitly specified language primitive.
4. The conceptual level contains primitives with definite cognitive interpretation, corresponding to conceptual meaning which is language independent.
5. The linguistic level contains primitives of linguistic terms of nouns and verbs, which is language dependent.

Table 1.1 Knowledge representation formalisms (Guarino 1995)

Level	Type	Primitives	Interpretation	Main feature
1	Logical	Predicates, functions	Arbitrary	Formalization
2	Epistemological	Structuring relations	Arbitrary	Structure
3	Ontological	Ontological relations	Constrained	Meaning
4	Conceptual	Conceptual relations	Subjective	Conceptualization
5	Linguistic	Linguistic terms	Subjective	Language dependency

A visualized form of knowledge representation such as graph-based form is highly adapted to model knowledge (Chein and Mugnier 2008). Conceptual Graph is an example of graph-based knowledge representation introduced by Sowa in 1976 (Sowa 1976, 1984). Graph-based approach to knowledge modeling has the advantage of easy understanding by human. Since a graph is easy to be visualized on screen and to be understood by human, it takes advantage for control and maintenance, also for human verification and validation. Logic defined on graph-based knowledge benefits computational processing and calculation. From many researchers' opinion (Sowa 2000, Artale 1996, Guarino 1998, Harmelen et al. 2008), ontology is a relevant logical and graphical model for knowledge representation, and sometime it is also said to be a category or classification system to represent knowledge.

1.3 What Is Ontology?

"Ontology" originates from philosophy, and it has been growing into popular research in computer science and information system. In the philosopher's perspective, for examples, Aristotle and Kant, ontology is the study of existence. It refers to a system of categories to describe the existence of the real word, or the classification of being. Although Aristotle's ontology has been developed for more than two thousand years, his classification system is still relevant for defining nowadays ontological classification systems. Table 1.2 shows the Aristotle's ten categories (Aristotle, Categories, 1990) to express things or existence:

Table 1.2 Aristotle's categories

	Categories	Descriptions
1	Substance	What, or being
2	Quantity	How much
3	Quality	What kind
4	Relation	With respect to
5	Place	Where
6	Time	When
7	Position	To lie
8	State	To have
9	Action	To do
10	Affection	To undergo

Table 1.3 Kant's categories

	Categories	Sub-categories	Descriptions
1	Quantity	Unity	Universal
		Plurality	Particular
		Totality	Singular
2	Quality	Reality	Affirmative
		Negation	Negative
		Limitation	Infinite
3	Relation	Inherence and Subsistence	Categorical
		Causality and Dependence	Hypothetical
		Community	Disjunctive
4	Modality	Possibility or Impossibility	Problematical
		Existence or Non-Existence	Assertoric
		Necessity or Contingence	Apoditic

Immanuel Kant presented a new successful categories system in 1781. The system is divided into four categories and further into three sub-categories in each main category as shown in Table 1.3. This classification system and categories are also relevant to nowadays ontology development, especially the ontology is highly dependent on relation that describes an entity or a being. The sub-categories of this reference can be seen as different types of object, properties and relation of ontology.

Ontology is a fundamental form of knowledge representation about the real world. In the computer science perspective, ontology defines a set of representational primitives with which to model a domain of knowledge or discourse (Gruber 2008). The representational primitives of the ontology contain classes, attributes (properties) and relationships between classes. They are used to model knowledge of particular application domains.

Ontology sometimes is regarded as for conceptual analysis and domain modeling (Guarino 1998). It is used to analyze the meaning of an object in the world, of a particular domain, and provides a formal specification to describe the object. The object is being "conceptualized" in this case. Gruber (1992) provided a very short definition about ontology – "An ontology is a specification of conceptualization". The formal specification is in support of some sort of knowledge representation model, being generated, analyzed, and processed by computer. The conceptualization has been defined in AI researches (Genesereth and Nilsson 1987, Nilsson 1991) as a structure of $<D, R>$. The structure defines D as a domain and R as a set of relations on the domain D. This suggests that ontology and conceptualization process are created as domain dependent and relational based.

Ontology aids the development of knowledge-based system, enabling knowledge sharing and reuse. It enables intelligent communication between computers, such as the communication language used in software agents (Lee 2007). Formalized specification allows knowledge engineers to develop their own ontology by reusing and sharing with each other.

1.4 Ontology Modeling in Computer Systems

Ontology modeling in computer system, called computational ontology, is rather simpler than that in philosophy. It provides a symbolic representation of knowledge objects, classes of objects, properties of objects, and the relationships among objects to explicitly represent knowledge about an application domain. The ontology modeling is usually simplified into different kinds of mathematical definition, logical definition, or structural language.

1.4.1 Computational Ontology Representation

Computational ontology is generally represented in different kinds of abstraction: top-level ontologies, lexical ontologies and domain ontologies. They create conceptualization by defining vocabularies. The vocabularies are organized by formal relationships to create dependent linkages. Some of them are organized as a tree structure and some of them are in graph.

1.4.2 Top-Level Ontologies

Top-level ontologies (also known as upper ontologies) are limited to concepts that are universal, generic, abstract and philosophical. They are generic enough to deal with high-level abstraction and broad ranges of domain areas. Concepts defined in a top-level ontology are not specific to a particular domain (e.g. medical, science, financial), but it provides only a structure and a set of general concepts from which a domain ontology can be constructed. This top-level ontology promotes data interoperability, semantic information retrieval, automated reasoning and natural language processing.

The standard upper ontology working group (IEEE SUO WG 2003) develops a standard of upper ontology for computer application for data interoperability, information search and retrieval, natural language process (NLP), etc. Examples of existing upper ontologies include SUMO (suggested upper merged ontology) (SUMO Ontology 2004), the CYC ontology (OpenCyc 2003), and also SUO 4D ontology (SUO 4D Ontology 2005).

The SUMO has been proposed as a starter document for the SUO working group. It creates a hierarchy of top-level things as "Entities", and subsumes "Physical" and "Abstract". SUMO divides the ontology definition into three levels: the upper ontology (the SUMO itself), the mid-level ontology (MILO), and the bottom-level domain ontology. Mid-level ontology serves as a bridge between the upper abstraction and the bottom-level rich details of domain ontologies. Beside the upper and mid-level ontology, SUMO also defines rich details of domain ontologies including Communications, Countries and Regions, distributed computing, Economy, Finance, engineering components, Geography, Government, Military (general, devices, processes, people), North American Industrial Classification System, People, physical elements, Transnational Issues, Transportation, Viruses, World Airports A-K, World Airports L-Z, WMD (SUMO Ontology 2004).

```
Physical
    Object
        SelfConnectedObject
            ContinuousObject
            CorpuscularObject
        Collection
    Process
Abstract
    SetClass
        Relation
    Proposition
    Quantity
        Number
        PhysicalQuantity
    Attribute
```

Fig. 1.1 SUMO top level

The OpenCyc is also upper-level ontology. It is some formalized common knowledge, and it models the general knowledge base and intended to solve commonsense problems. The entire Cyc ontology contains hundreds of thousands of terms with relationship among the terms to model human consensus reality. It contains a knowledge server to serve for its Cyc knowledge base, an inference engine, and it also defines CyCL representation language for knowledge representation. It is an upper-ontology available for defining some lower level ontology knowledge such as domain specific knowledge, and domain specific facts and data as shown in the Figure 1.2 (OpenCyc 2003).

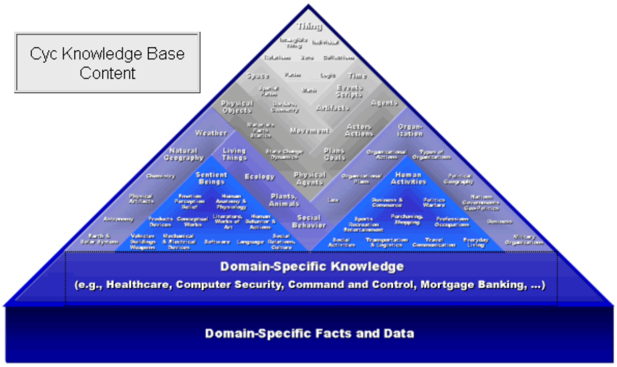


Fig. 1.2 OpenCyc upper-level ontology hierarchy (OpenCyc 2003)

1.4.3 *Lexical Ontologies*

A lexical ontology is an ontology describing linguistic knowledge, and it tries to model the word meaning by ontological structure. Examples of this type of ontology are WordNet (Miller 1998), and HowNet (Dong and Dong 1998). WordNet is an English based system which organizes words on the basis of lexical taxonomical semantic relationships, but the usage on WordNet is strictly limited to English based application only. HowNet is a Chinese-English bilingual lexical ontology describing the semantic relationship between concepts and the relationship between the attributes of concepts. It covers over 65,000 concepts in Chinese that are equivalent to about 75,000 concepts in English. Lexical ontology is useful for developing knowledge based system that may requires text analysis such as word sense disambiguation, word sense similarity calculation, words sense annotation and ontological annotation.

WordNet

WordNet is originally designed as a lexical database of English word (Miller 1998). It could be used as a lexical ontology to represent knowledge for computer text analysis and artificial intelligence application development, especially for many natural language related applications. Word-Net defines synsets to group English nouns, verbs, adjectives and adverb into sets of synonyms, and uses different grammatical rules of distinguishes between them (noun verbs, adjectives and adverb). It is helpful to model concept of words and its semantic relationship. It has been used for various natural language text analyses such as word sense calculation and disambiguation.

WordNet research has been extended to ImageNet (Deng et al. 2009), a large-scale hierarchical image database. It uses the meaningful concept in WordNet connecting to image data. This is a practical example of using Word-Net as knowledge to build an intelligent information system, a concept-based image database.

HowNet

HowNet is lexical database of Chinese word developed by Dong (1998). It is a common-sense knowledge based for modeling inter-conceptual relations and inter-attribute relations of Chinese lexicons concepts and their English equivalents (Dong and Dong 1998, HowNet 2003). How-Net is aimed for Chinese language processing by using its constructed knowledge based of Chinese words. Similar to the synsets of WordNet, HowNet defines its specific Sememe-Network to model the inter-conceptual relations between Chinese lexicons concepts. HowNet is a fully computable electronic database. Knowledge of HowNet is structured by a graph. A graph based example is shown in Figure 1.3 for describing different concepts, property, attributes, and their inter-relationship. HowNet also defines a taxonomy (Figure 1.3) which serves as the upper ontology to model category for Chinese lexical concepts.

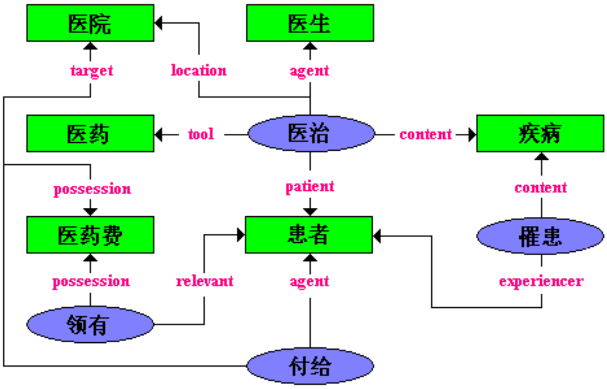


Fig. 1.3 Graphical expression of Chinese lexical concept in HowNet

Although there have been a lot of works done in these lexical ontologies, and they have also conceptualized lexical knowledge quite effectively, the main problem of these ontologies is that they are manually created in the entire process. A drawback of manual process is hard for maintenance, such as adding new knowledge, revising and updating existing knowledge. The concept and usage of words are changing all the time, so the defined knowledge of words is not permanently valid. Therefore, a continuous manually updating work and re-construction are required and thus make the process ineffective.

```
- {thing|万物} {entity|实体:{ExistAppear|存现:existent={~}}}

   - {physical|物质} {thing|万物:HostOf={Appearance|外观},
          {perception|感知:content={~}}}

      - {animate|生物} {physical|物质:HostOf={Age|年龄},
              {alive|活着:experiencer={~}},{die|死:experiencer={~}},
          {metabolize|代谢: experiencer={~}},
          {reproduce|生殖:agent={~},PatientProduct={~}}}

         - {AnimalHuman|动物} {animate|生物:HostOf={Sex|性别},
                  {AlterLocation|变空间位置:agent={~}},{StateMental|精神
              状态:experiencer={~}}}

            - {human|人} {AnimalHuman|动物:HostOf={Name|姓名}
               {Wisdom|智慧}{Ability|能力},
               {think|思考:agent={~}},{speak|说:agent={~}}}
```

Fig. 1.4 Basic data - Taxonomy in HowNet

1.4.4 Domain Ontologies

A domain ontology is tied to a specific domain which can be extended from upper ontology. It should be defined for specific domains because even some huge ontology like Cyc, contains over ten thousands of concepts modeling the generic and high-level concepts, but is still not deep enough to express the conceptual and low-level of a specific domain (a domain such as medical, science, financial, etc.). In order to model a domain knowledge and make the information expressive and understandable by machines, domain ontology is developed based on the concept formation in the particular domain of interest. Domain ontology is preferably built based on an available upper ontology (e.g. SUMO, Cyc) for the ease of mapping and integration between different domain ontologies created by different specialists or researchers, as to enhance sharing and usability.

Unlike upper ontology which is usually built for reasoning commonsense knowledge, domain ontology is mainly built for reasoning a specific domain of knowledge. The domain ontology is boarder and more general for defining knowledge. In another words, domain ontology is less abstract but more specific. It is therefore more useful to build intelligent application because computer application is usually developed for particular target domains. Figure 1.5 shows an ontology tree specified for entertainment, a part of entertainment domain.

Most of the application ontologies are domain dependent, but they are shared among each other crossing over different domains. Ontology engineering usually aims to define and create domain ontologies rather than top-level ontologies and lexical ontologies, and the recent researches on domain ontology engineering will be reviewed in Chapter 2.

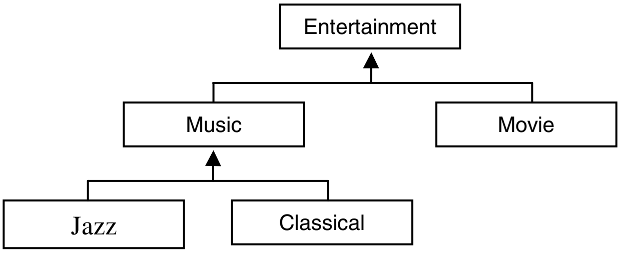


Fig. 1.5 Ontology sample of entertainment domain

Chapter 2
Ontology Engineering

Abstract. Ontology engineering is the research to find out methods of creating ontology knowledge in computer system. This research concerns about formalizing the structure of ontology, and the development of some algorithms for automatic or semi-automatic ontology learning. Some of the work involves developing tools for ontology engineer or domain expert to create ontology manually. This chapter introduces the concept of ontology engineering, some related tools and also some related technology that is useful to the development of ontology learning algorithms.

2.1 Introduction

Ontology becomes more important in A.I. research, as ontology is a recognized relevant knowledge representation for building intelligent knowledge based systems. Ontology engineering is the research of developing theories, methods, and software tools that help creating and maintaining ontologies. In most of these researches, the ontology refers to domain ontology rather than upper or lexical ontology. This is because domain ontology is a rich and core knowledge to build domain dependent applications while upper and lexical knowledge is mostly for general use. The most challenge of ontology engineering is how to create and maintain ontology. Various approaches adopted by knowledge engineers to create ontology in the past include: 1. manual ontology creation, 2. automatic ontology creation, and 3. semi-automatic ontology creation.

Manual ontology creation approach – The simplest way to create ontology is by manual process. After defining the structure of ontology specification, domain experts start implementing the data of ontology that conforms to the specification. It requires a large human workforce to create domain ontologies from scratch. Different domain experts create ontologies by different and subjective views. The ontology knowledge is thus not exact and may not be relevant to all knowledge domains. Maintaining and updating such ontology is also time consuming and inefficient as every process requires deep analysis by human.

Automatic ontology creation approach – As manual ontology creation is not a practical approach in ontology engineering, many researches try to develop an automatic process for the ontology creation. We understand that there are rich knowledge in many kinds of computer data such as text documents and database

E.H.Y. Lim, J.N.K. Liu, and R.S.T. Lee: Knowledge Seeker, ISRL 8, pp. 13 – 25.
springerlink.com

records. Researchers want to develop an automatic process to extract knowledge of a specific format (ontology) from those computer data. However, it is always unsuccessful due to inaccurate and low-quality ontology created by the automatic process. Therefore, semi-automatic ontology creation is the most practical approach in nowadays ontology engineering research.

Semi-automatic ontology creation approach – It adopts the automatic ontology creation process, in addition to the involvement of human works in the ontology creation. It is always developed with an ontology engineering framework, human efforts are taken consideration into the process of the framework, such as ontology refinement, validation and verification. This is coordination between human and machine automation to find out an optimal process for ontology engineering. Before the ontology engineering process is successfully automated, a semi-automatic process is proven to be the most practical approach to create high quality ontology.

2.2 Ontology Fundamentals

Ontology engineering defines not only the ontology engineering method, but also the formal structure and specification for representing the ontology. The specification defines how the format of the ontology is formalized, for both machine and human processing.

2.2.1 Ontological Structure

Ontology in computer system defines how knowledge of the real word is formalized by computer logics. It contains different components to comprise a whole ontology. An ontological structure is to define how those components gather and construct together to represent a valid ontology.

A 5-tuple based structure (Maedche 2002) is a commonly used formal description to describe the concepts and their relationships in a domain. The 5-tuple core ontology structure is defined as:

$$S=(C, R, H, rel, A)$$

Where:

- C is the set of concepts describing objects.
- R is a set of relation types.
- H is a set of taxonomy relationship of C.
- rel is a set relationship of C with relation type R, where $rel \subseteq C \times C$
- A is a set of description of logic sentences

rel is defined as a set of 3-tuple relations: $rel = (s, r, o)$, standing for the relationship of subject-relation-object, where s is the subject element from C, r is the relation element from R, and o is the object element from C. In this 5-tuple ontological structure, knowledge is mainly represented by the logic sentences A, and the most important component is rel where it defines 3-tuple based concept relationship.

A graphical representation of the 3-tuple structure is shown in Figure 2.1, in which subject *s* is being defined as a node of source n_1, object *o* is being defined as a node of target n_2, and relation *r* is being defined as the association link between n_1 and n_2.

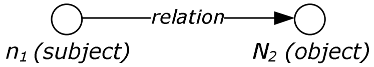


Fig. 2.1 Subject-relation-object representation

So we should define the main component in the ontological structure, which is called concept *(C)*. A concept is an ontological object, material or non-material, spatial or non-spatial. A name (or label) is provided to represent a concept (and also relation). A sign is also a valid description to define a concept. A sign is used to signify a concept which is a definition from Semiology, a sign system (Guiraud 1975). A name (or sign), in lexical level, called a lexicon, is used to declare a concept. The lexicon structure is defined as:

$$L=(G_C, G_R, ref_C, ref_R)$$

Where

- G_C is a set of named elements called a lexicon entry for concepts *C*
- G_R is a set of named elements called a lexicon entry for relations *R*
- ref_C is a set of lexicon references for concepts *C,* where $ref_C \subseteq G_C \times C$
- ref_R is a set of lexicon references for relations *R,* where $ref_R \subseteq G_R \times R$

In this lexicon structure, a concept can be referenced by a single or multiple lexicons, and also a single lexicon can be referred to multiple concepts. An ontology structure is therefore defined by both the core ontology structure *S* and lexicon structure *L*: *O=(S, L).*

2.2.2 Ontological Taxonomy

Ontology structure contains taxonomy relations and non-taxonomy relations. The taxonomy relation of concepts forms a hierarchical tree-based structure to model ontology knowledge. Taxonomy is a type of classification system, classifying objects with parent-child relationship. The difference between ontological relations and non-taxonomical relations would be like (Figure 2.2): “Jazz” and “Classical” are the subtype of “Music” (taxonomical relation). While a classical music of India is another relation between “Classical” and “India” (non-taxonomical relation).

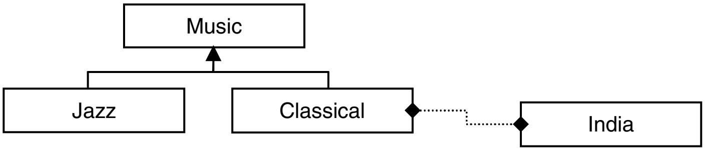


Fig. 2.2 Taxonomical relations and non-taxonomical relations in ontology

The taxonomy is ontology in the form of a hierarchy. It forms a “is a” relation, such that child is-a type of parent (subtype). In the example shown in Figure 2.2, Jazz is a type of Music, India is a Country of Asia. Taxonomy is found everywhere in information system, as simple as web directory, news category, Topic Maps (ISO/IEC SC 34/WG3 2008), etc.

2.2.3 Ontological Relations

Ontological relations are more than taxonomy relations. Many ontologies definition contains parent-child relation (taxonomy) and also part-whole relation (partonomy). A taxonomy divides a concept into species of kinds (e.g. “Jazz and “Classical” are types of “Music”), while a partonomy divides concept as a whole into different parts (e.g. “India” and “China” are parts of “Asia”). Ontology includes both taxonomy and partonomy relations as shown in Figure 2.3.

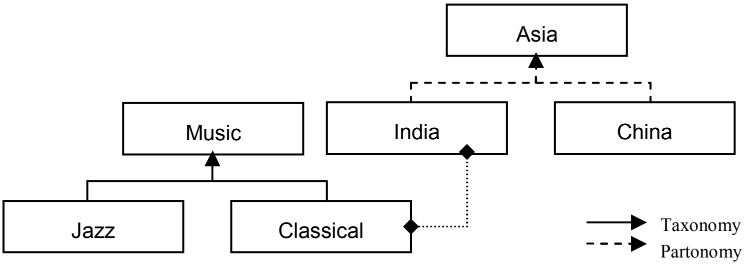


Fig. 2.3 Taxonomy and partonomy relations in ontology

Combining taxonomy and partonomy relations creates transitive taxonomy relations for the concept like: “Indian classical music” is a type of “Asian music”, but not “Indian classical music” is a part of “Asian music”. Another example is given in Figure 2.4: “Dog leg” is a type of “Animal leg”

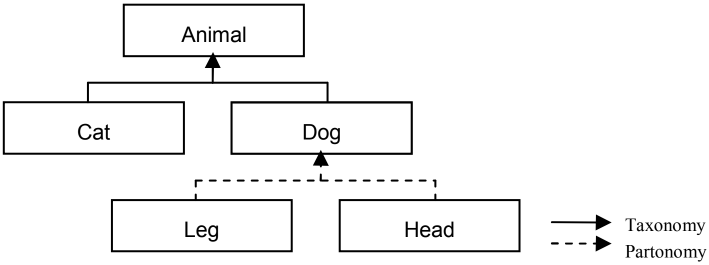


Fig. 2.4 Taxonomy and partonomy relations in ontology

From the view of top-level ontology, ontological relations can be distinguished into three types according to the nature of universals or particulars of the object itself (Schwarz and Smith 2008):

1. Universal to universal – the related objects are both universal. For example: both universals "*Animal*" and "*Dog*" form a parent-child ("*is a*") relation. Such that: "*Dog*" is a (an) "*Animal*". Another example: both universals "India" and "Asia" from a part-whole ("*is part of*") relation, such that "*India*" is a part of "*Asia*".
2. Instance to universal – one related object is universal while the other one is a particular. It is a type of instantiation relations between two objects. For example: a particular dog named "*Lucky*" from an instantiation relation to the universal "*Dog*", such that "*Lucky*" is a "*Dog*".
3. Instance to instance – the related objects are both particulars. For example, the particulars "*Lucky's Leg*" and "*Lucky*" from a part-whole ("*is part of*") relation in the level of instances, such that "*Lucky's Leg*" is a part of "*Lucky*".

A graphical representation of the above examples is given in Figure 2.5.

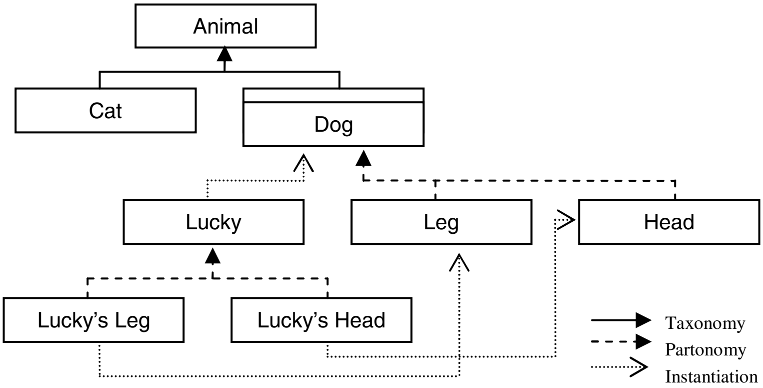


Fig. 2.5 Ontological relations by universals and particulars example

Figure 2.6 shows another example of the ontological relations between universals and particulars: Universal to universal – both universals "India" and "Asia" from a part-whole ("*is part of*") relation, such that "*India*" is a part of "*Asia*". Instance to universal – a particular country named "*India*" from an instantiation relation to the universal "*Country*", such that "*India*" is a (an) "*Asia Country*". Instance to instance – a particular "*Asia Country*" from a parent-child ("*is a*") relation in the level of instances, such that "*Asian Country*" is a "*Country".*

There are more relations in the figure, which are neither taxonomy, partonomy nor instantiation. They are the relations between "*India*" and "*Classical*", "*Asia*" and "*Asia Country*". These ontological relations should be named specifically, for example: "India" produces "Classical Music", or "Classical Music" produced in "India". "*Asian Country*" resides in "*Asia*", or "*Asia*" consists of "*Asian Country*". This conforms to the subject-relation-object pattern as described in ontological structure, as "*India*" (subject) "*produces*" (relation) "*Classical Music*" (object), or "*Asian Country*" (subject) "*resides in*" (relation) "*Asia*" object.

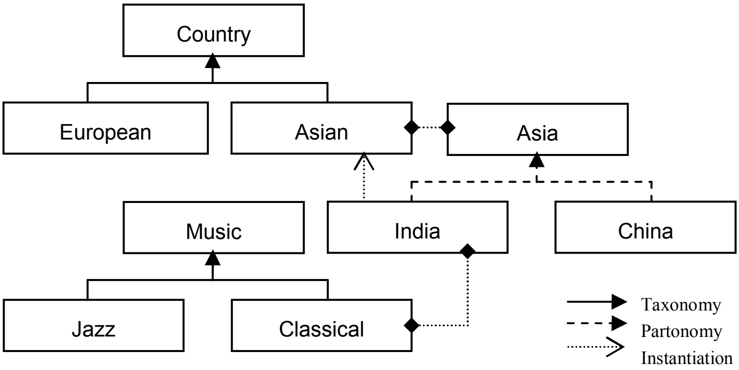


Fig. 2.6 Ontological relations by universals and particulars example

The logic of relations defines the inclusion of class membership, and the instantiation transitive relations, so that:

- Assume *A* is a *B*: if *x* is instance of *A*, then *x* is instance of *B*
  "*Lucky*" is an instance of "*Dog*", so "*Lucky*" is an instance of "*Animal*"
- Assume *A* is a part of *B*: if *x* is instance of *A*, *x* is not instance of *B*
  "*Lucky's Leg*" is an instance of "*Leg*", but "*Lucky's Leg*" is not an instance of "*Dog*"
- Assume *A* is a *B*: if *x* is related to *A*, then *x* is related to *B*
  "*Asia*" consists of "*Asian Country*", so "*Asia*" consists of "*Country*"
- Assume *A* is part of *B*: if *x* is related to *A*, then *x* may be related to *B*
  "*Classical Music*" produced in "*India*", so "*Classical Music*" produced in "*Asia*"

Such that the transitive relations of instantiation and other named relations are summarized in Table 2.1. This table defines that the "is a" relationship has inheritance characteristic so that if a concept is a sub-type of some other concept, that concept inherits all the relations properties of the super-type one, while the "part of" relationship has that inheritance characteristic for only some situations:

Table 2.1 Transitive relations characteristic

relation	is a	part of	instantiation	others
is a	+	+	+	+
part of	–	+	–	+ / –

2.2.4 Lexical Relations

Existing lexical databases such as WordNet and HowNet define semantic relations between lexicons. They are typical lexical relations that can be transformed to an ontology structure. WordNet defines different types of ontological primitives and lexical relations, and these lexical relations are able to map on to ontology (Maedche 2002). The matching of lexical relations in WordNet to ontological such as taxonomy (hyponymy), partonomy (meronymy), and antonymy can be figured out:

Lexical Hierarchy

WordNet uses synsets, sets of semantic relation – synonym, to organize lexical information in meaning. Hyponym in WordNet is an important semantic relation that organizes lexical terms into meaning. The hyponymy system is a lexical hierarchy organizing nouns in relations of subordination (subsumption or class inclusion). The lexical hierarchy formed by nouns in WordNet is an inheritance system so that taxonomy can be built correspondently. Definitions of these relations in WordNet are:

- Hypernym – defines a whole class of specific instances: W_1 is a hypernym of W_2 means that W_2 is a kind of W_1.
- Hyponym – defines a member of a class: W_1 is a hyponym of W_2 means that W_1 is a kind of W_2.

Parts and Meronymy

The part-whole relation between nouns is also defined as a semantic relation called meronymy in WordNet. This relation has an inverse such that if W_1 is a meronym of W_2 then W_2 is said to be holonym of W_1. Hyponyms can inherit meronyms features, that mean if W_1 and W_2 are meronyms of W_3, and W_4 is a hyponym of W_3, then W_1 and W_2 are also meronyms of W_4 by inheritance, like the transitive feature in ontological relation. Definitions of these relations in WordNet are:

- Meronym – defines the member of something: W_1 is meronym of W_2 if W_1 is a part of W_2.
- Holonym – defines a whole-part relation: W_1 is a holonym of W_2 if W_2 is a part of W_1.

Antonymy

Antonymy is semantic opposition of nouns defined in WordNet. Deadjectival nouns are the most common nouns for this opposition, such as the words "*happiness*" and "*unhappiness*". Typical example also includes the words "*men*" and "*women*", being defined that W_1 is antonymy of W_2 if W_1 is not W_2.

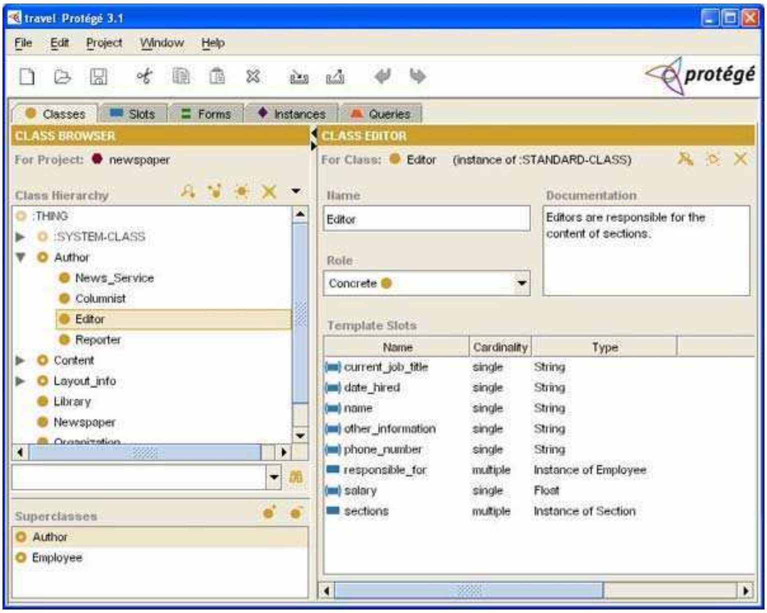
2.3 Ontology Engineering Tools

Ontology engineering tool or an ontology editor is an application that helps ontology engineers and domain experts to create and maintain ontology. These applications manipulate ontology in different kinds of ontology languages, providing on screen ontology management functions such as creation, edit, verification, import, export, etc. Examples of such applications include Protégé, Onto-Builder, OntoEdit, Construct, etc (Denny 2009).

2.3.1 Protégé Ontology Editor

Protégé is a free and open source ontology editor for creating domain model and knowledge based information system with ontologies (Protégé 2009). It is being developed by Stanford University. Protégé editor supports various ontology languages such as RDF, RDF(S), DAML+OIL, XML, OWL, Clips, UML, etc. Two main methods of modeling ontologies support in Protégé include:

- A frame-based ontology modeling is in accordance to Open Knowledge Base Connectivity (OKBC 1995). This model consists of a set of classes representing domain concepts in subsumption hierarchy (ontological taxonomy), a set of associated slots in classes representing the properties and relationships (ontological relations), and a set of instances of those classes (instantiation). See the screen of this modeling method in Protégé (Figure 2.7).
- An OWL (OWL 2004) based ontology modeling for the Semantic Web (W3C Semantic Web 2009) (see Chapter 4). OWL is a language describing classes, properties and their instances. It formally specifies how to derive the logical consequences of the ontology, and aimed for developing Semantic Web applications. See the screen of this modeling method in Protégé (Figures 2.7 and 2.8).



For Project: ● newspaper

Class Hierarchy

:THING
:SYSTEM-CLASS
Author
News_Service
Columnist
Editor
Reporter
Content
Layout_info
Library
Newspaper
Organization

Superclasses

Author
Employee

For Class: ● Editor (instance of :STANDARD-CLASS)

Name: Editor

Documentation: Editors are responsible for the content of sections.

Role: Concrete ●

Template Slots

Name	Cardinality	Type
current_job_title	single	String
date_hired	single	String
name	single	String
other_information	single	String
phone_number	single	String
responsible_for	multiple	Instance of Employee
salary	single	Float
sections	multiple	Instance of Section

Fig. 2.7 The Protégé-Frames editor for OKBC ontology modeling

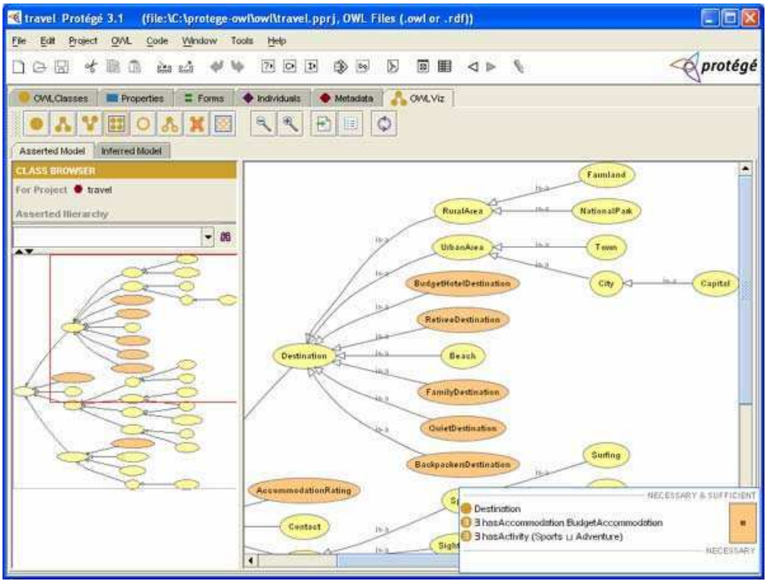


Fig. 2.8 The Protégé-OWL editor for OWL ontology modeling

2.4 Ontology Learning from Text

Text is the most direct resource of human knowledge. Human beings write texts about what they view, what they know, and what they think about the world, so they are descriptive data that enable human to share and exchange their knowledge. Although analyzing textual data by computer is not an easy task, many methodologies on ontology learning from text have been widely developed in recent years (Maedche 2002, Buitelaar and Cimiano 2008). Most of them use artificial intelligent approaches to develop the methodologies, and the automatic text learning process is the goal of these researches. They use many artificial intelligent approaches such as information retrieval, machine learning, natural language processing, statistical mathematics, etc. to build the ontology learning system. However, the ontology learning outcome is sometimes not satisfactory to represent human knowledge. This is because computational ontology is defined explicitly, but knowledge in textual data is vague and implicit. There are difficulties to convert an implicit knowledge from text to a formalized ontology representation, in terms of both its quantity and quality. Quantity refers to that the ontology learning outcome is not comprehensive enough to express the whole knowledge domain, and they should have missed out some useful knowledge inside the text. Quality refers to that the ontology learning outcome cannot express the knowledge relevantly. In other words, the formalized knowledge from automatic learning process is partly irrelevant or wrongly generated.

In view of knowledge engineering, automatic ontology learning from text is very helpful in formalizing ontology knowledge. With the use of automatic ontology learning method, the ontology can serve for two main purposes: First, the ontology outcome can improve the performance of traditional information system by increasing the intelligent ability with embedded basic ontology knowledge. Although the embedded ontology is incomplete for the entire knowledge domain, it is still relevant to enhance the performance by intelligence. Second, the ontology outcome can serve as an intermediate ontology, or a base ontology for human to further develop and revise it. The incomplete ontology or an ontology with unsatisfied quality can aid human to develop a desired ontology for the knowledge domain, as they are not required to build the entire ontology from scratch.

The degree of knowledge formalization describes different steps in ontology learning process. It defines different levels of knowledge data to be extracted step-by-step from text learning method. In brief, the extraction process should be started from the raw text data (text document in natural language text) to the final ontology knowledge representation. The degree of knowledge formalization contains seven different levels for learning unstructured textual content to ontology and logical rules (Navigli and Velardi 2008), as represented in Figure 2.9.

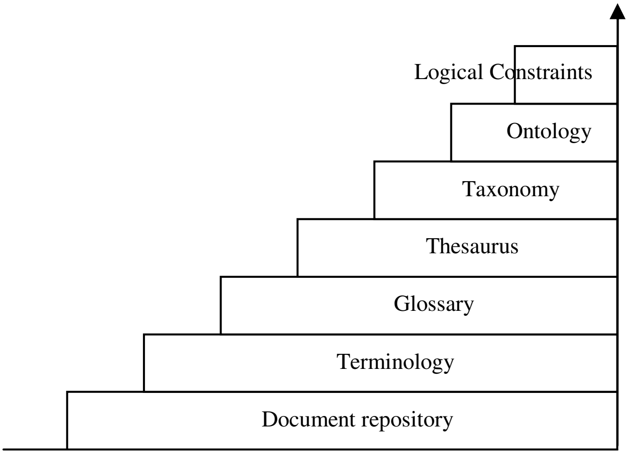


Fig. 2.9 Degree of knowledge formalization in seven levels

Common ontology learning processes include five main steps in cycles (Meadche 2001), they are:

1. Extraction – to extract ontology components such as lexicons, terminologies, glossaries, taxonomies, and ontological relations from text sources.
2. Pruning – the ontology which created from extraction, import and reuse is pruned to fit its primary purpose.
3. Refinement – a human process made to the pruned ontology as to complete the ontology at a fine granularity.
4. Application – application of the ontology serves as a measurement and validation of the ontology

5. Import and reuse – Import, reuse and merging existing ontology by mapping result. This is to refine and update the existing ontology by new learning result.

The domain ontology learning process from text mainly consists of the steps of learning terminology (terminology extraction or glossary extraction) and learning taxonomies (taxonomical relations extraction).

2.4.1 Learning Terminology

Terminology refers to a set of words or word phases that specifically used in a domain text. The automatic terminology extraction has widely been discussed in some past researches (Velardi et al. 2007, Park et al. 2002, Navigli and Velardi 2004). The methods focus on the combination of natural language processing techniques, statistical measurement, and text mining with pattern matching algorithms (Kalvans and Muresan 2001). Most of these automatic terminology or glossary extraction methods work at the lexical level which refers to the term (a word or word phase) that used in a domain text.

Terminology can be described technically by a graph-theoretic object. The graph object consists of nodes associated together by links, and the whole structure indexed by version number (Smith et al. 2006). The common components of terminologies *n* are defined as a triple:

$$n =< p, S_p, d >$$

- A preferred lexical term *t* signed for the terminology
- Any synonyms S_p which the term may have
- A definition *d* for the term and its synonyms

A formal definition of terminology can be provided by description logic as to record information about terminologies in graph-theoretic object. The named nodes in the terminology graph are defined respectively as n_1, n_2, n_3,…. The links associated to the named nodes are defined as L_1, L_2,…. Different versions of the terminology are defined as v_1, v_2,…. The terminology definition is then an ordered triple:

$$T =< N, L, v_n >$$

where *N* is a set of nodes n_1, n_2, n_3,… for every *n* is a triple of $< p, S_p, d >$ with *p* as a preferred lexical term., S_p as a set of synonyms, and *d* as the definition. The link is defined as *L* in which it consists of an ordered pair $<r, L_r>$, which includes a relation *r* and an ordered pairs of preferred lexical terms that defined as $L_r = <p, q>$. Finally the v_n is a versions number.

Extracting lexical terms by frequency measurement

A typical information retrieval approach to extract a lexical term is the *tfidf* measurement (Oddy 1981). The *tfidf* measurement is used to measure a term importance, for extracting important and relevant lexical terms in a document corpus *D*. The $tfidf_{l,d}$ of the lexical term *t* for the document *d* is defined as

$$tfidf_{t,d} = tf_{t,d} * \log\left(\frac{|D|}{df_t}\right)$$

where $tf_{t,d}$ is the frequency of occurrence of the lexical term *t* in a document *d*, and the df_t is the overall document frequency containing the lexical term *t*. By the *tfidf* measurement, terms that appear too rarely or too frequently are weighted lower than an average, aimed to filter irrelevant terms. The *tfidf* value for a lexical term *t* is defined as:

$$tfidf_t := \sum_{d \in D} tfidf_{t,d} \ , \quad tfidf_t \in \Re$$

where *D* is a document corpus and $\Re$ is a threshold that defined to filter out irrelevant lexical terms.

Learning Domain terminology by probabilistic measurement

Learning terminology basically relies on analyzing a domain classified text corpus. A high frequency term in a corpus is identified as a terminology. OntoLearn (Navigli and Velardi 2004) uses two probabilistic measurements called Domain Relevance (*DR*) and Domain Consensus (*DC*) respectively to measure terminology terms for domains. The suggested *DR* is a quantitative measurement of the amount of information captured within a target domain corpus with respect to a larger collection of corpora. For example, given a corpus with *n* classified domains $\{D_1, D_2, \ldots, D_n\}$, the Domain Relevance (*DR*) is defined as

$$DR_{t,k} = \frac{P(t \mid D_k)}{\max_{1 \le j \le n} P(t \mid D_j)}$$

where $P(t \mid D_k)$ is a conditional probabilities estimated as:

$$E(P(t \mid D_k)) = \frac{f_{t,k}}{\sum_{t' \in D_k} f_{t',k}}$$

where $f_{t,k}$ is the observed frequency of the term *t* in the domain D_k documents in the corpus. The extracted terms by *DR* measurement is assigned by the second measurement *DC*. The *DC* is a second analysis taking into account not only the overall observed frequency of a term in a corpus, but also its existence in a single document. The term *t* in documents $d \in D_k$ is measured by a stochastic variable estimation throughout all $d \in D_k$. The Domain Consensus (*DC*) is defined as:

$$DC_{t,k} = \sum_{d \in D_k} \left(P_t(d) \log \frac{1}{p_t(d)} \right)$$

where

$$E(P_t(d_j)) = \frac{f_{t,j}}{\sum_{d_j \in D_k} f_{t,j}}$$

All non-domain candidate terms are thus filtered by combining measurement of *DR* and *DC:*

$$\alpha DR_{t,k} + \beta DC_{t,k}$$

where $\alpha, \beta \in (0,1)$. This probabilistic measure provides a practical approach for extracting domain terminologies based on lexical term.

2.4.2 Learning Taxonomy

Domain taxonomy learning refers to three-step processes: terminology extraction, glossary extraction and taxonomical relations extraction. Taxonomy learning can be grouped into three main areas (Velardi et al. 2007):

1. Methods based on manually and automatically created regular expressions applied to text documents.
2. Methods based on statistical measurement of terms extracted from text documents.
3. Methods based on dictionary parsing.

All of the three methods have some drawbacks. First, the method based on regular expression is a simple lexical pattern created for matching on the text documents. The pattern matching is created by phrase like *"is a"*, *"is called*", *"is a type of*", etc. Creating these patterns is time consuming and error prone, and it does not guarantee the quality of matching results. Language dependency is also a concern for creating the lexical pattern. Second, the method based on statistical measurement is mostly based on the comparison and analysis of the contextual features of terms, such as hierarchical clustering algorithm (Cimiano et al. 2004). This automatic analysis method creates a taxonomy result that is difficult for human understating. As a result, it is difficult for doing evaluation by a human judgment since all the kind-of relations are learnt by statistical measurement, including noise and idiosyncratic data. Finally, the method based on dictionary parsing is highly relying on human constructed dictionary, disadvantages such as circularity of definitions and overgenerality has been discussed in past research (Ide and Véronis 1994).

Chapter 3
Text Information Retrieval

Abstract. Text information retrieval is the most important function in text based information system. They are used to develop search engines, content management systems (CMS), including some text classification and clustering features. Many technologies about text information retrieval are well developed in the past research. This chapter reviews those information retrieval technologies and some related algorithms which are useful for further development into ontology learning method.

3.1 Information Retrieval Model

With the rapid growth of Internet technologies, huge amount of web information are now available online. Information retrieval (IR) on web is so becoming a very important research area. Most of the web documents are created in the form of unstructured or semi-structured text. Traditional IR on text data including text classification, text clustering, and text-based search engines are mostly processed on keyword-based. Keyword-based text retrieval model gives inaccurate result in many IR systems and also lacks of intelligence features. Intelligent IR system applies computational knowledge model, or computational ontology, to enhance the retrieval algorithms. Intelligent IR systems improve the performance, in terms of its accuracy, over traditional IR systems to gain effective result in nowadays information environment.

There are three common traditional information retrieval applications: content searching, text classification/clustering, and content management. Most of these use statistical or machines learning approaches such as *tf-idf,* support vector machine (SVM), *k-NN*, neural network, and fuzzy set system to support text analysis in many application developments.

3.1.1 Term Weighting Model

The common approach of text information retrieval is to represent text document content by sets of content identifiers (or terms). Term importance is the main measurement in this approach as every single term may have different importance (weight) to the information domain. Documents in this model are thus represented by a collection of weighted terms. For example, a given document $d_{j,}$ is

E.H.Y. Lim, J.N.K. Liu, and R.S.T. Lee: Knowledge Seeker, ISRL 8, pp. 27 – 36.
springerlink.com

represented by a collection of terms $T = <t_1, t_2, \ldots, t_m>$ where t_i represents the importance values, or weight, of term i assigned to document d_j.

The term weighting system varies among different approaches, but is mostly based on counting the term frequency in a document. For example, a collection of n documents indexed by m terms are presented by an m x n term by document matrix $A=[a_{ij}]$. For each a_{ij} in the matrix A is defined as the observed (or weighted) frequency of term i which occurs in document j. Table 3.1 and Figure 3.1 show an example of term-by-document matrix for $m = 8$ and $n = 7$.

Table 3.1 Content of terms and documents

Documents	Terms occurrence
d_1	t_6
d_2	t_1, t_2, t_5
d_3	t_2, t_5, t_8
d_4	t_1, t_4, t_6
d_5	t_1, t_7
d_6	t_3, t_7
d_7	t_1, t_3

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Fig. 3.1 8 x 7 term-by-document matrix for 7 documents and 8 terms

3.1.2 Text Classification

A text classification system refers to constructing a classifier in which, given a set of classes $C=\{c_1, c_2, \ldots, c_i\}$ and a document d, find out the most relevant class c_i of which the document d belongs to. The classifier is a function $f_i(d)\rightarrow\{0,1\}$ expressing the relevancy value of the document d for the class c_i. A classical text classification model consists of documents as the input, process with natural language processing, feature extraction, feature weighting, feature reduction, classification engine, and then being classified into relevant classes or categories. Here we review three common classification approaches for the classification engine, they are: 1. Statistical classification, 2. Functional classification, and 3. Neural classification.

Statistical classification

A typical algorithm from IR for classification is the Rocchio algorithm (Rocchio 1971). It is based on statistical measurement technique and the vector space model (VSM) with TF/IDF weighting, where *tf-idf* is defined as:

$$tfidf_{t,d} = tf_{t,d} * \log\left(\frac{|D|}{df_t}\right)$$

where $tf_{t,d}$ is the frequency of occurrence of a lexical term t in a document d, and the df_t is the overall document frequency containing the lexical term t. In this approach, the semantic of document is represented by a collection of lexical terms occurring in it. In addition, the weighting is normalized by cosine normalization for adjusting the weights to fall in the [0,1] interval, so that every document is represented by a vector of equal length:

$$w_{t,d} = \frac{tfidf_{t,d}}{\sqrt{\sum_{t \in T}\left(tfidf_{t,d}\right)^2}}$$

Another typical probabilistic technique for text classification is the Naïve Bayesian classification. It measures the probability that a document d belongs to a class c_i,, where the d is represented by a vector of terms $\{t_1,t_2,\ldots,t_n\}$. This is described by the conditional distribution:

$$p(c_i \mid t_1, t_2, \ldots, t_n) = \frac{p(t_1, t_2, \ldots, t_n \mid c_i)\, p(c_i)}{\sum_{c \in C}\left(p(t_1, t_2, \ldots, t_n \mid c)\, p(c)\right)}$$

where $p(c_i)$ denotes the probability of any document belonging to the class c_i, and the left side is the conditional probability of the document with a vector of terms $\{t_1,t_2,\ldots,t_n\}$ that belongs to class c_i. Assuming that the order of term occurrences is independent from the classification, the conditional probability can be computed as:

$$p(t_1, t_2, \ldots, t_n \mid c_i) = \prod_{j=1,n} p(t_j \mid c_i)$$

Functional classification

In functional classification, every document is represented as a dot in a multidimensional space, where the size of the dimensional space is equal to the size of term number. Some simple and effective functional classifications include the k-Nearest-Neighbors (kNN) and support vector machines (SVM).

kNN (Kwon & Lee, 2003) approach measures the similarity or distances between documents. When all documents are represented as a dot in a multidimensional space, kNN considers the *k*-nearest (most similar) neighbors to the new documents (Figure 3.2). The document is classified to the class if all the *k*-nearest neighbors belong to that same class. Otherwise, if all the *k*-nearest neighbors do not belong to a same class, the document is classified to the largest group of classes of the neighbors.

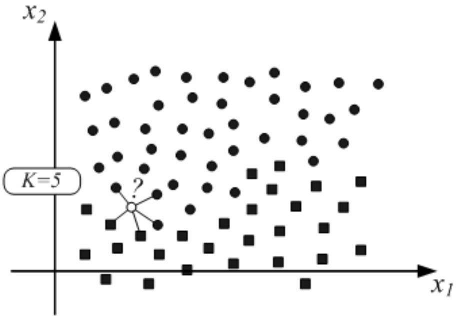


Fig. 3.2 kNN classification approach

Neural classification

Neural classification uses the technique of artificial neural network (ANN) as its classification model. The ANN is an electronic network of "neurons" based on the neural structure of the human brain. Neurons consist of nodes and links. Input and output values are composed of nodes, while weights composed of links (Figure 3.3).

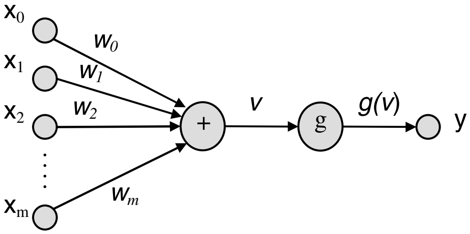


Fig. 3.3 Inputs, output and function in a neuron

- x_i is the set of input values
- w_i is the associates weights of the inputs
- *g* is the function of the sums of weights, and it maps the results to output
- *y* is the output value

The neurons are organized into multi-layer to form a multi-layer perception (MLP) neural network, for example, a 3-layer neural network as shown in Figure 3.4. The neural network uses a feed forward, back propagation (BP) method to do the classification.

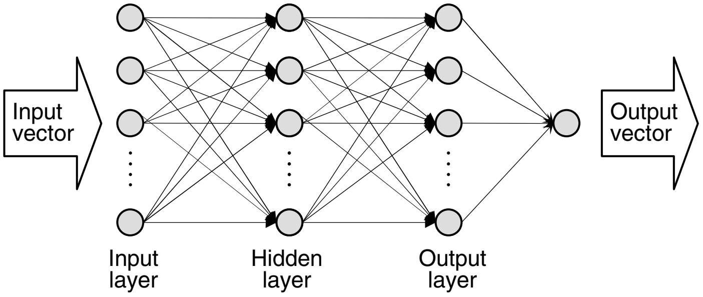


Fig. 3.4 3-layer structure of neural network classification

3.1.3 Text Clustering

Text clustering is the process of grouping text documents to its related classes of topic area. Traditional and popular algorithms include single-link and complete-link hierarchical methods, K-means partition methods, Rocchio TFIDF methods, and Support Vector Machines (SVM) methods. Most of these methods operate on similarity measurement. This is to measure the similarity between two feature vectors in common feature space, let's denote the two feature vectors: $\vec{x} = (x_1, ..., x_m)$ and $\vec{y} = (y_1, ..., y_m)$. The widely used similarity functions include Euclidean distance, known as *L2* norm:

$$L_2(\vec{x}, \vec{y}) = \left(\sum_{i=1}^{m} (x_i - y_i)^2 \right)^{1/2}$$

Another similarity measure, measuring the similarity between two vectors by finding the cosine of the angle between them, is known as cosSim (cosine similarity):

$$\cos Sim(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{\left\| \vec{x} \right\| \left\| \vec{y} \right\|}$$

Hierarchical methods use distance function between clusters. K-means method depends on the notion of a cluster centroid. Centroid is defined as *u(C)* of a cluster *C* which is the mean of the group of points that forms the cluster:

$$\vec{u}(C) = \frac{1}{|C|} \sum_{\vec{x} \in C} \vec{x}$$

3.2 Feature Selection and Reduction

Text classification and clustering always involve high dimensional space. The problem of high dimensionality is the increase in the number of terms, which will increase the time for processing and also increase noises for the classification and task. Feature selection and reduction are therefore important processes to decrease the dimensional space. By using feature selection, text classification uses smaller number of terms for processing. The main challenge of feature selection process is how to measure the importance of terms and the optimum number of terms selected or filtered, in order to attain a higher classification or clustering performance.

Feature selection can improve the accuracy and efficiency of text classification by filtering irrelevant and redundant terms from the corpus (Yang and Pederson 1997). Common feature reduction techniques are principal component analysis (PCA) (Lam and Lee 1999) and latent semantic indexing (Sebastiani 2002). PCA (Duda et al. 2001, Wang and Nie 2003) maximizes the total scatter in all classes and result in retention of non-discriminative information (Busagala et al 2008). Canonical discriminative analysis (CDA) can be applied to acquire more discriminative information. PCA is applied on a set of training corpus. From the term weighting model, for *N* documents in the training corpus $D = \{T_1, T_2, ..., T_N\}$ with *n*-dimensional term space, each text document is represented by a feature vector *T* which is defined as:

$$T = \{t_1, t_2, ..., t_n\}^T$$

where *n* is the term size (dimensionality) and t_i represents the term frequency of term *i* occurring in document D_i . T is the transpose of the vector.

In principal component analysis, total covariance matrix $\sum$ of the training documents corpus $D = \{T_1, T_2, ..., T_N\}$ is defined as:

$$\frac{1}{N}\sum_{T \in C}(T - M)(T - M)^T$$

where *M* is the total mean vector of the training document corpus and is defined as:

$$M = \frac{1}{N}\sum_{T \in C} T$$

The eigenvalues and eigenvectors of training sample are defined as:

$$\sum \Phi_i = \lambda_i \Phi_i (i = 1,2,...,n)$$

Feature vectors are then reduced and obtained by selecting *m* principal components from the following definition and thus high-dimensional space is reduced to the size of *m:*

$$\Phi_i^T T(i = 1,2,..., m)$$

Other common feature selection methods including Information Gain (IG) (Quinlan 1986) and Chi-square statistics are able to improve text clustering performance (Liu et al. 2003). In IG measurement, the number of bits of information obtained for a category prediction is calculated by observing the presence or absence of a term in a document. The information gain of term *t* to a category *c* is defined as:

$$IG(t, c_i) = \sum_{c \in \{c_i, \neg c_i\}} \sum_{t' \in \{t, \neg t\}} P(t', c) \cdot \log \frac{P(t', c)}{P(t') \cdot P(c)}$$

In Chi-square (χ^2) statistical measurement, features (terms) that have high dependency on a category can be selected. χ^2 works on measuring the dependency degree of a term *t* from a particular category *c*. A two-way term-to-category contingency table (Table 3.2) is filled up with the observed term frequency $O_{i,j}$ where $i \in \{t, \neg t\}$ and $j \in \{c, \neg c\}$. Therefore, $O_{w,c}$ is the observed frequency (number) of documents in category *c* which contains the term *t;* $O_{t, \neg c}$ is the observed frequency of documents which are not in category *c* and contains the term *t*; $O_{\neg t,c}$ is the observed frequency of documents which are in category *c* and do not contain the term *t*; and $O_{\neg t, \neg c}$ is the observed frequency of documents which are neither in category *c* nor contain the term *t*.

Table 3.2 Term-to-category contingency table

	c	$\neg c$	Σ
t	$O_{t,c}$	$O_{t, \neg c}$	$O_{t,c} + O_{t, \neg c}$
$\neg t$	$O_{\neg t,c}$	$O_{\neg t, \neg c}$	$O_{\neg t,c} + O_{\neg t, \neg c}$
Σ	$O_{t,c} + O_{\neg t,c}$	$O_{t, \neg c} + O_{\neg t, \neg c}$	$O_{t,c} + O_{t, \neg c} + O_{\neg t,c} + O_{\neg t, \neg c} = N$

The observed frequency is compared to the expected frequency $E_{i,j}$ where $i \in \{t, \neg t\}$ and $j \in \{c, \neg c\}$. $E_{i,j}$ and χ^2 for term *t* and category *c* are defined as:

$$E_{i,j} = \frac{\sum_{a \in \{t, \neg t\}} O_{a,j} \sum_{b \in \{c, \neg c\}} O_{i,b}}{N}$$

$$\chi^2_{t,c} = \sum_{i \in \{t, \neg t\}} \sum_{j \in \{c \neg c\}} \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}}$$

Basically, features can be selected based on the higher χ^2 values, where lower χ^2 values are recognized as irrelevant term to the category, and hence reduced the size of the feature space.

3.3 Search Engine Modeling

Search engine is a practical application that uses the information retrieval (IR) techniques for large amount of text collections. Search engines are developed for various applications like desktop file search, enterprise search and the most obvious application – web search engine.

Search engine consists of two major processes: they are the indexing process and querying process. The indexing process further breaks down into the process of text acquisition, text transformation, and index creation (Croft et al 2010). The indexing process takes text documents as inputs, and then creates a document index as output. The document index is a sort of indexed terms or features of document in the document database. The querying process is then further broken down into the process of querying transformation, an IR model, and ranking. The querying process takes the document index as input, goes for the querying process, and then creates retrieval document as the result (Figure 3.5).

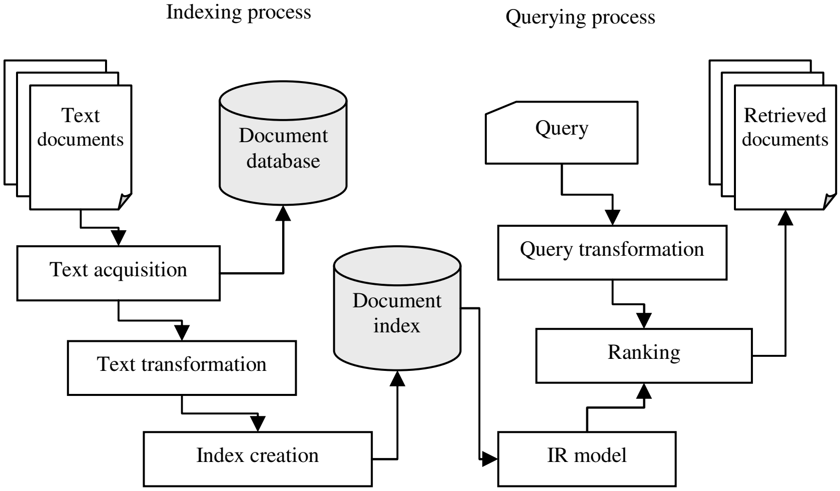


Fig. 3.5 Search engine process and its components

A web search engine gathers web documents (HTML) as the input source of indexing process. It therefore requires a process of capturing (or crawling) the web documents to form the document database. Web mining technique is always used for the process. The process usually supports automatic crawling (web mining robot), and is often preprocessed by some text analysis such as automatic document classification, and clustering will also be done before the indexing process.

Web Mining

Traditional data mining is also known as knowledge discovery (KDD) in database. This is an emerging research that using computational power to discovery knowledge and interested rules from a large database. It provides more efficient means of analyzing data in scientific or business application we are unable to handle the otherwise. Data mining is a semi-automatic process to discovery knowledge, rules, pattern in data store, and is implemented with several technologies such as artificial intelligence, machine learning, computational statistics, etc.

In web mining, the entire World Wide Web is treated as the large database for mining the interested knowledge or data. The data source is retrieved from different web servers connected to the Internet, and the data might be in any format but mostly HTML. There are different tasks of web mining. First, web structure mining is the process of extracting links and the organization in single or in collaborating with other web sites. It is used to find out the web site structure in order to get all web files (HTML) from the target web server. Second, web usage mining is similar to the data mining approach in processing web log file data, which can automatically extract patterns of user accessing a particular web page. Web logs or user logs are the main data source for this web mining task. This is aimed to extract more knowledge, useful data and information about the users. And last, web content mining is to analyze the content of the extracted web files (mostly HTML). This web content mining process deals with semi-structured HTML and try extract text contents and related information. This content is used for further process such as used for the search engine indexing. Web mining is therefore a necessary process for building a web search engine, for collecting the web documents as the source documents.

3.4 Evaluation Methods

Error rate is the most practical measurement to evaluate the information retrieval model. This measurement is aimed to calculate the retrieval accuracy, in terms of precision, recall, and f-measure. It is done by first observing the retrieval correctness from the result, as shown in Table 3.3:

Table 3.3 The table of retrieval result

	Relevant	Irrelevant
Retrieved	*TP*	*FN*
Not retrieved	*FP*	*TN*

- *TP*(True Positive) – the number of relevant documents, retrieved as relevant
- *FP* (False Positive) – the number of relevant documents, not retrieved as relevant
- *FN* (False Negative) – the number of not relevant documents, retrieved as not relevant
- *TN*(True Negative) – the number of not relevant documents, not retrieved as relevant.

3.4.1 Performance Measurement

Precision – It measures the accuracy of the retrieval model, by calculating the percentage of correctly retrieved documents to the whole retrieved result set. It is defined by:

$$precision = \frac{TP}{TP + FP}$$

Recall – It measures the ability of the retrieval model to retrieve correct documents from the whole data set, by calculating the percentage of correctly retrieved documents to all the documents that should be retrieved. It is defined by:

$$recall = \frac{TP}{TP + FN}$$

F-measure – It measures the harmonic average of precision and recall. It is defined by:

$$f - measure = \frac{2 \times precision \times recall}{precision + recall}$$

Chapter 4
Web Data Semantics

Abstract. The current web system is largely built on HTML. HTML is originally designed for human consumption only. Therefore they are not designed to "understand" the web content on their own. Enriching web contents with semantic data is aimed to solve this problem. Semantic web is a kind of this technology which adds more structural markup data to the semi-structured information in HTML page. This semantic markup data gain benefits in machines understandability. Therefore it can enhance agent application to process web content. There is also close relationship between ontology and semantic web as ontology is the key elements for building up of semantic web content.

4.1 Semantic Web

The semantic web is designed for not only providing web data for human uses, but also creating the data that the machines can process. The main vision of semantic web is to create machines-processable data and define how machines act to the data and make a web system become more intelligent.

We need the semantic web because web information is overloaded nowadays. Since the amount of web data is too much for human consumption, we need machines to help us to do a lots of information processing before it deliver to us. This information processing such as information filtering, searching, and recommendations require high machine-intelligence, the technologies of semantic web enable us to development such kinds of intelligent system efficiently. The adoptions of semantic web technologies can benefits many organizations in their current business process and improve its efficiency. Daconta et al (2003) described some ideas of what a semantic web can utilize the greatest benefits of organizations, and it is revised as shown in Figure 4.1.

Decision Support and Marketing

The semantic web consists of knowledge as its core component, and it is in machine-processable data that enable a machine to analyze and return certain useful result to uses. These analytical results can aid for decision making and marketing purposes. Machines can even give a certain expert advice or recommendation to the user, provide more valuable knowledge to aids decision making.

E.H.Y. Lim, J.N.K. Liu, and R.S.T. Lee: Knowledge Seeker, ISRL 8, pp. 37 – 46.
springerlink.com

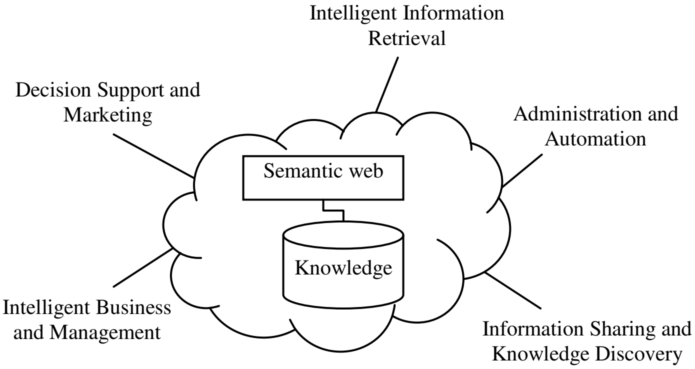

Intelligent Information Retrieval
Decision Support and Marketing
Administration and Automation
Semantic web
Knowledge
Intelligent Business and Management
Information Sharing and Knowledge Discovery

Fig. 4.1 Classification model and its sub-processes

Intelligent Business and Management

Traditional business applications such as e-commerce and customer relationship management provide only static data, such as product information, transaction records, customer information, etc. Semantic web has reasoning ability that can be used in matchmaking for e-business. It helps in associating potential customers with business partners or sales components. This intelligent business and management features create more opportunities for profit in an organization.

Information Sharing and Knowledge Discovery

Traditional information systems store data in its proprietary database and it is not designed for sharing and reuse in other systems. Even with the data export feature of one system may create difficulties for using and understanding it in another system. However, the semantic web technologies define data (knowledge) not only machine-processable, but also application-independent. That means the data (knowledge) can be easily exchanged, shared, and reused in other systems for processing. This application-independent data provides information sharing ability and also enhances the knowledge discovery feature. This is because different systems react with the data differently and so it may derive new knowledge by its reasoning logics and its own knowledge.

Administration and Automation

With the features of machine-processable data, information sharing, and intelligent reasoning, a lot of administration and tasks automation can be developed for e-business solutions. The automated tasks may include: finding a certain product on Internet, processing with the buying task, booking air tickets and hotels, negotiating the price, searching for a good restaurant, etc. These automated tasks can be developed in an intelligent agent model, so that the agent can operate on behalf of its host (user) to complete the desired task automatically and intelligently without human intervention.

Intelligent Information Retrieval

Traditional information retrieval systems such as classification system and search engine rely on keyword data, because they are not embedded with any processable knowledge. This kind of systems is inefficient because keyword based information retrieval task is lacking of high precision, in the sense that it always return invalid results that do not match users' need. Semantic web technologies overcome this problem by developing a knowledge-based information retrieval system. A knowledge-based information retrieval task always returns more accurate result than a keyword-based information retrieval task. Therefore the semantic web can handle an information retrieval task with more intelligence.

4.1.1 W3C Semantic Web

Existing web technologies rely on HTML. HTML is originally designed for human consumption only. The problem of the existing web architectures is that machines are unable to process. The semantic web is designed to solve this problem, by enriching web content with markup data. This markup data means to add more structural information to the semi-structured information in HTML page. This markup data gain benefits in machines understandability. Therefore it can enhance agent application to process web content. There is also close relationship between ontology and semantic web as ontology is the key element for building up semantic web content. This section describes the semantic web defined by W3C (W3C semantic web 2007), which is about the underlying concepts and technologies supported for developing a semantic web.

The Semantic Web Stack

Figure 4.2 visualizes the semantic web stack by W3C. It is separated into different layers that enable to develop a semantic web. Starting from the bottom layer, the self-describable markup language, XML, is being used, it enables data exchange across the web, but it does not represent any meaning and knowledge embedded in the data. So RDF (Resource Description Framework) and RDF schema are defined and to be built on top of XML, it can be used to model the abstract representation of data-meaning and data-semantics, this data-semantic in RDF (based on XML) hence can be easily processed and understood by software agent. Finally, the ontology knowledge is modeled in OWL. OWL defines more detail about properties, classes, relationship between classes, cardinality, equality, etc. SPARQL defines the query language for semantic web data. This comprises the lower layers (data layer) in the semantic web stack. The upper layer of semantic web architecture consists of proof and trust. It describes the issues of accessibility and credibility of the distributed data. Web application could do reasoning about the confidence of the derived result based on these layers.

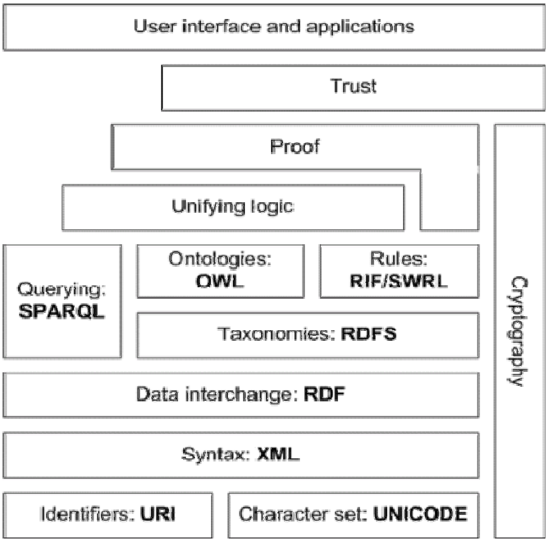


Fig. 4.2 Semantic Web stack

4.2 Semantic Modeling

Semantic modeling in information technologies refer to mapping or formalizing human knowledge to some kinds of language syntax (Allemang and Hendler 2008). Human knowledge are usually expressed in unstructured natural language which is very difficult for computer processing, therefore we need some structured language syntax to model the underlying "semantics" behind the natural language. The main idea of semantic modeling is to associate a term in a statement with a concept in the real world that the term refers to. Various technologies have been developed to handle the semantic modeling task. According to the ability to express the knowledge, we simplify those semantic modeling techniques in the levels from weak to strong semantics (Daconta et al 2003), as shown in Figure 4.3.

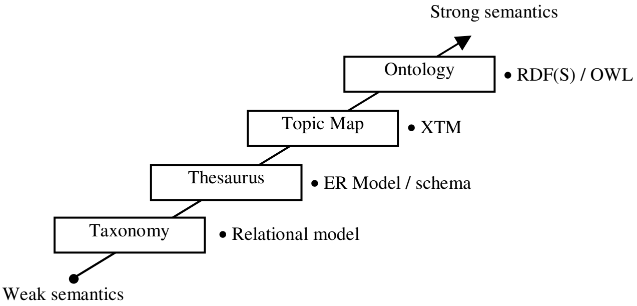
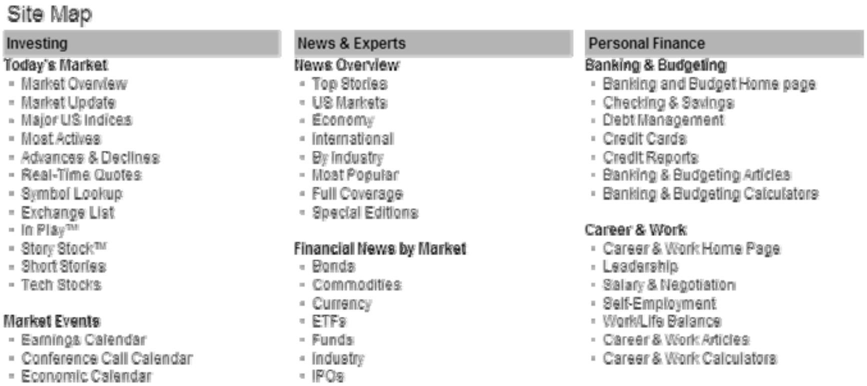


Fig. 4.3 The semantic modeling techniques in levels

4.2.1 Taxonomy

Taxonomy describes knowledge in hierarchical structure or in the semantics of the parent/child relationship. Taxonomy is a type of classification system in the form of class and sub-class relation. A typical taxonomy is the animal classification in biology. For example, animal is classified into chordata, arthropoda, mollusca, annelidia, etc, and chordate is further classified into aves, reptilian, amphibian, mammalia etc, and mammalia contains human, cat, dog, etc. Taxonomy is useful in describing living things in the real world and it has had a profound role in biology for a long time. Taxonomy can also be found everywhere in information technology environment, such as the folder structure in a computer drive, and the "site map" of a web site. For example, the content of a finance web site can be classified into investing, news & experts, personal finance, etc, and investing can be further classified into today's market, market event etc. and finally today's market contains the hyper links to market overview, market update, etc. (Figure 4.4).



Site Map

Investing

Today's Market
- Market Overview
- Market Update
- Major US Indices
- Most Actives
- Advances & Declines
- Real-Time Quotes
- Symbol Lookup
- Exchange List
- In Play™
- Story Stock™
- Short Stories
- Tech Stocks

Market Events
- Earnings Calendar
- Conference Call Calendar
- Economic Calendar

News & Experts

News Overview
- Top Stories
- US Markets
- Economy
- International
- By Industry
- Most Popular
- Full Coverage
- Special Editions

Financial News by Market
- Bonds
- Commodities
- Currency
- ETFs
- Funds
- Industry
- IPOs

Personal Finance

Banking & Budgeting
- Banking and Budget Home page
- Checking & Savings
- Debt Management
- Credit Cards
- Credit Reports
- Banking & Budgeting Articles
- Banking & Budgeting Calculators

Career & Work
- Career & Work Home Page
- Leadership
- Salary & Negotiation
- Self-Employment
- Work/Life Balance
- Career & Work Articles
- Career & Work Calculators

Fig. 4.4 Example of a financial site map organized in a taxonomy hierarchy

4.2.2 Thesaurus

Thesaurus can be defined as "controlled vocabulary arranged in a known order and structured so that equivalence, homographic, hierarchical, and associative relationship among terms are displayed clearly and identified by standardized relationship indicators" (ANSI/NISA Z39.19-1993 [R1998], p.1). Therefore, it describes knowledge more than the taxonomy. The relationships among terms in a controlled vocabulary are used to associate the meaning of a term with the meaning of other terms. WordNet is an example of thesaurus for English and HowNet for Chinese. Table 4.1 shows different types of semantic relations and their examples:

Table 4.1 Examples of semantic relations in thesaurus

Relationship Type	Example
Equivalence	
Synonymy	"HK" / "Hong Kong"
Homographic	
Homonym	"Mouse" (*animal*) / "Mouse" (*input device*)
Hierarchical	
Hypernym	"Mouse" / "Mammal" (*child-of*)
Hyponym	"Mammal" / "Mouse" (*parent-of*)
Meronym	"Window" / "House" (*part-of*)
Holonym	"House" / "Window" (*has-part*)
Associative	
Cause-effect	"Accident" / "Injury"
Attribute-host	"Color" / "Cloth"
Material-product	"Grapes" / "Wine"
Location-event	"Hospital" / "Medical treatment"
Event-role	"Medical treatment" / "Patient"

4.2.3 Topic Maps

Topic Maps is an ISO international standard for the representation of structured information model. It is a kind of semantic web technology, and it is used to represent the relationships between abstract concepts and information resources. Topic Maps model can be therefore divided into two separated spaces: 1. Topic space – consists of topics that represent concepts in the real world, and 2. Resource space – consists of resource files that are electronic files such as web pages, text documents, multimedia files, etc. Topics related together by association connection to form concepts and it relates resource file by occurrence connection (Figure 4.5). Topic maps can be expressed in XTM file syntax (Figure 4.6).

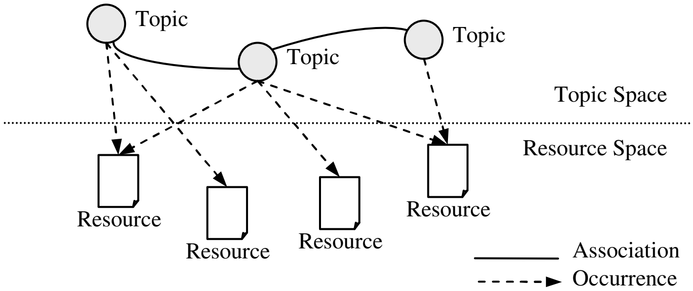


Fig. 4.5 The semantic modeling techniques in levels

Components in Topic Maps

- Topics – a machine-processable format to represent anything about electronic resources, or non-electronic resource (or real world things such as people, places, events, etc.).
- Associations – used to represent the relationship between topics to form concepts.
- Occurrences – used to represent or refer to a resource about a concept formed by topics.

```
<topic id="ax">
  <baseName>
       <baseNameString>AX Finance Inc.</baseNameString>
  </baseName>
  <occurrence>
       <resourceRef xlink:href="http://www.axfinance.com/"/>
  </occurrence>
</topic>
```

Fig. 4.6 XTM example

4.2.4 Ontology

Ontology is the strongest semantic modeling techniques among the other techniques discussed above. The word ontology is borrowed from philosophy. In computer science, an ontology precisely defines a term about a specific domain, represents an area of knowledge, and standardizes the meaning. According to Gruber (1993), "an ontology is an explicit specification of a conceptualization". Ontology usually consists of a set of vocabulary (concepts), taxonomy, relationships, properties, etc. Therefore, it can model an area of knowledge in a stronger "semantic" sense than the taxonomy and thesaurus. Table 4.2 presents the components of an ontology.

Table 4.2 Components of an ontology

Component	Description
Classes	Set of concepts that describe objects
Instances	Particular things of objects
Relationships	Associations about meaning among those things
Properties	Property values of those things
Functions	Functions and processes describing those things
Constraints	Description logic and rules describing those things

4.2.5 Ontology Languages for the Semantic Web

Ontology language is the markup language which can be used to model the data semantic architecture in the data layer of the Semantic Web architectures. The language available to markup the ontology and data semantic for semantic web includes XML, RDF, RDFS, DAML + OIL and OWL.

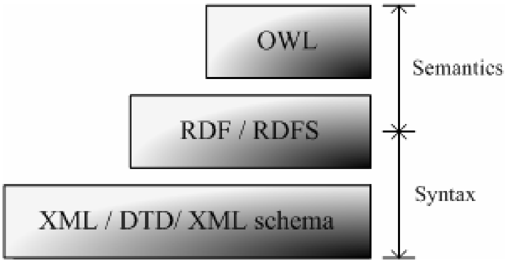


Fig. 4.7 Language for ontology modeling

Extensible Markup Language XML

XML is the most basic markup language for data exchange between machines. It is structured format to enable processing by machines. XML with specific DTD or XML schema specifies the syntactic conventions, but the required data semantics are not defined in XML data and therefore upper markup language is required to build on top of XML.

Resource Description Framework (RDF)

A language framework by W3C recommendation has defined the meta-data description of web-based resource. RDF presents data in subject-predicate-object triple written as *P (S, O)*, and can be visualized by a labeled edge between two nodes as shown in Figure 4.8. This triple notation allows object playing the role of a value, which enables the chaining of two labeled edges in a graphical visualization, as shown in Figure 4.9.

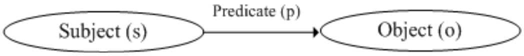


Fig. 4.8 RDF subject-predicate-object triple

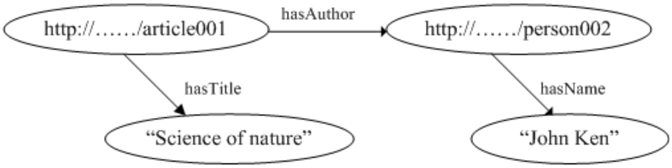


Fig. 4.9 RDF triple relations

The RDF triples *P (S, O)* is defined as: *hasAuthor(article001, person002)*, *hasTitle(article001, "Science of nature")*, *hasName(person002, "John Ken")* which can be serialized in RDF/XML syntax as shown in Figure 4.10.

```
<rdf:Description about="http://domain/article001">
     <hasAuthor rdf:resource="http://domain/person002"/>
            <hasName rdf:resource="John Ken"/>
     </hashAuthor>
     <hasTitle rdf:resoource="Science of nature"/>
<rdf:Descrtiption>
```

Fig. 4.10 RDF example

RDFS (RDF-Schema)

RDF schema is used to describe the meaning of data in RDF, providing additional facts to RDF instance. Machines process RDF by mapping RDF data information from one RDFS from one to another. RDFS allows ontology developer to define a set of vocabularies for RDF data (e.g. *hasAuthor*, *hasTitle*, *hasName*) and specify the types of object with these properties that can be applied to, thus it is defined the relationship exists between two things (an existence). It also models the class-subclass, property-subproperty relationship that is common in an ontology model, defining the generalization-hierarchies of the properties and classes used in RDF data.

Web Ontology Language (OWL)

OWL provides greater machines readability of web content compared to RDF and RDFS, by adding more vocabularies to describe properties and classes: such as relationship between classes (e.g. disjointness), cardinality (e.g. exactly one) which is not supported in RDFS. OWL therefore provides more expressive markup for ontology data for the semantic web. OWL is built on top of RDF and RDF Schema, and use the XML syntax of RDF (Figure 4.11). W3C Web ontology working group has defined OWL as three sublanguages: 1. OWL Full, 2. OWL DL, and 3. OWL Lite. Each sublanguage is defined for use by specific communities of implementer and users (W3C OWL 2004).

```
<owl:Ontology>
   <owl:Class rdf:about="#associateProfessor">
      <owl:disjointWith rdf:resource="#professor"/>
      <owl:disjointWith rdf:resource="#assistantProfessor"/>
   /owl:Class>
</owl:Ontology>
```

Fig. 4.11 OWL example

4.3 Semantic Annotation and Information Retrieval

Annotating web information in RDF/OWL meta-data is the key process for building up a Semantic Web. The annotation process (Handschuh & Staab 2003, Schreiber et al. 2004) requires combining the semantic content and data created by a large team of people. Semantic annotation process can be done manually or semi-automatically, CREAM (Handschuh & Staab 2003) is an example tool for building up annotation meta-data. However, using manual or semi-automatic annotation approaches assumes that the web information is static. Annotating dynamic source of web information requires fully automated annotation process, which is a more difficult task. Semantic annotation requires the ontology of the information domain (Soo et al. 2003). An annotation data is the context of the instantiation to ontology (instances of some classes that form the ontology) attached to or linked by an HTML document. HTML page deployed with an annotation data makes the information presented with semantic meaning and in more structured data format (such as RDF, OWL).

Traditional information retrieval systems focus on text-based retrieval and they are usually based on keyword matching. A problem of text-based retrieval system is that user might not have entered enough and explicit terms in their query. This is caused by many reasons such as users perhaps do not have complete knowledge of the domain, so that they usually cannot provide appropriate and exact keywords to construct a good query. Simple query expansion for finding more related terms in user query also suffers from creating too many unrelated terms, and thus reduces the precision in search result.

Semantic searching is the approach of searching information in more abstract “semantic” level instead of simply keyword matching (Gao et al. 2005). This can be done if the documents are well annotated with semantic meta-data (with various ontologies knowledge support). It requires more supports in the upper layer of the semantic web architecture to enable semantic searching. They are rules and logic, by which the search logics are defined for semantic matching, mapping and retrieval. While data are being annotated and stored underlay the top layer, which are data and ontology layers.

Part II

KnowledgeSeeker: An Ontology Modeling and Learning Framework

Chapter 5
Ontology Modeling Framework

Abstract. We have defined a knowledge representation model in KnowledgeSeeker called Ontology Graph, which is used to represent domain ontology and it can support ontological information search and management. The proposed Ontology Graph is a graphical based knowledge generated by semantic relations of Chinese words, and that semantic relations are formed by the ontology learning process automatically. This chapter first overviews the KnowledgeSeeker system and then presents the background idea and the implementation details of the proposed Ontology Graph.

5.1 KnowledgeSeeker – The System Overviews

KnowledgeSeeker is a comprehensive system framework which defines and implements the components of: 1. Ontology Modeling (the ontology structure), 2. Ontology Learning (the learning algorithm), 3. Ontology Generation (the format), and 4. Ontology Querying (the operations), as shown in Figure 5.1.

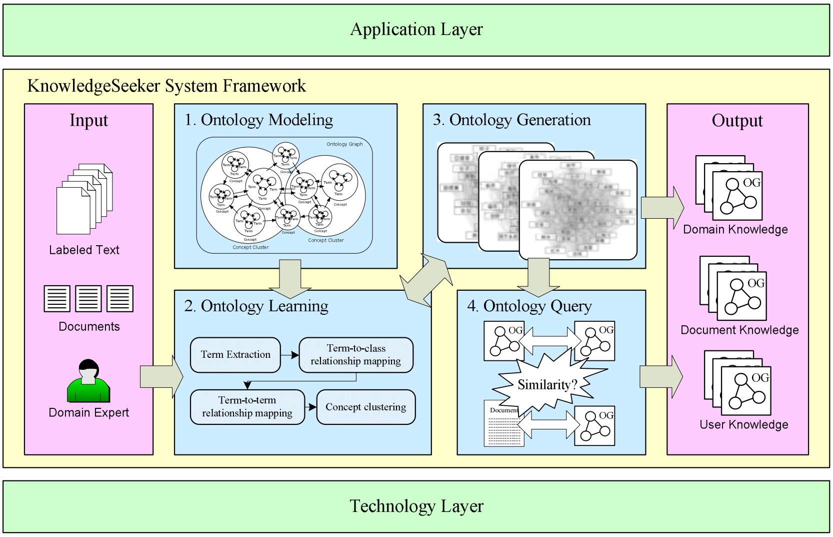


Fig. 5.1 Four modules in KnowledgeSeeker system framework

E.H.Y. Lim, J.N.K. Liu, and R.S.T. Lee: Knowledge Seeker, ISRL 8, pp. 49 – 70.
springerlink.com

The KnowledgeSeeker can be used to develop various ontology-based intelligent applications by using the four defined ontological components. These intelligent applications include such as knowledge-based information retrieval system, knowledge mining system, predication system, personalization system, intelligent agent system, etc. Therefore, the entire KnowledgeSeeker system framework breaks up into four modules for handling different kinds of ontological process:

Module 1 – Ontology Modeling

The ontology modeling module defines the conceptual structure that is used to represent the ontology data (knowledge) in the KnowledgeSeeker system. This is a kind of knowledge representation method and the knowledge is represented as Ontology Graph which will be described in the following of this chapter.

Module 2 – Ontology Learning

The ontology learning module concerns about the method of knowledge acquisition from texts. It defines the method of conceptualizing a domain of knowledge. The method is based on a statistical text learner, and the conceptualization process is about transforming knowledge of text into a machine-processable format, i.e. the defined Ontology Graph in Module 1. Figure 5.2 presents the knowledge components of ontology learning from text and the ontology learning module and its algorithm will be described in Chapter 6.

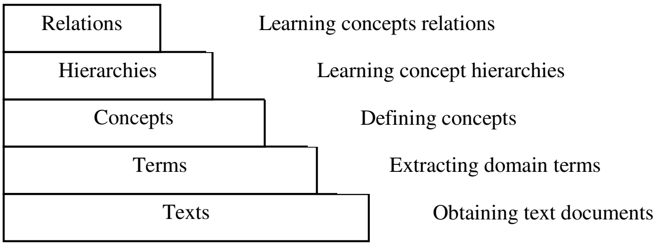


Fig. 5.2 Knowledge components of ontology learning from text

Module 3 – Ontology Generation

The ontology generation module formalizes the conceptual ontology model into a structural file format. The process uses a text corpus to generate domain ontologies in the form of Ontology Graph, and it visualizes the Ontology Graph in a graphical format. The ontology generation module and its definition will be described in Chapter 7.

Module 4 – Ontology Querying

The ontology querying module defines how system operates with Ontology Graphs. It is an important module that enables the use of KnowledgeSeeker system to develop various intelligent applications. The module defines operations such as Ontology Graph matching and querying that make the Ontology Graph data operable in developing various applications, such as text classification system, and text searching system. These applications can also be used to evaluate the performance of the querying methods, and the validness of the domain knowledge generated in the form of Ontology Graph. The ontology querying module will be described in Chapter 8.

5.2 Background of Signs System and Ontology

The sign system in semiotics or semiology, is the study of sign, languages, codes, sets of signals, etc. Some important features of sign had been proposed for Ontology development (Sowa 2000). *Concept* is the most important knowledge object in Ontology system, the very challenging issue in developing Ontology system is how to define *Concept*. Language (voice or text) can create *Concept*, and it is the most common communication methods used by human to express knowledge. This type of communication requires *Sign* (a voice, a visual object, or a word, etc.) for concept formation. The idea of using *Sign* for concept formation aids the development of Ontology system. This chapter introduces the sign system (semiotics and semiology), and how its features can be adopted to develop Ontology model in KnowledgeSeeker.

5.2.1 The Semiotics

Semiotics is the study of sign. It was first introduced by Peirce CS, a philosopher and logician. Semiotics concerns with finding meaning and representation of the real world things in many forms, and usually in the form of *text*. The term *text* in semiotics refers to a message which has been presented in some form, such as in audio (voice), video (visual), and writing (words). A sign in texts thus refers to a sound, an image, or a word, to form the medium of communication.

The semiotics is divided into three branches:

1. Syntax – the study of relations of signs among each other.
2. Semantics – the study of relations of signs to the things in the world, which the signs refer to.
3. Pragmatics – the study of relations of signs to those (people) who use them to refer to things in the world.

The Sign in Semiotics

The sign is a stimulus pattern that has a *meaning* (Port 2000). We make meaning by creating and interpreting the sign. There are three kinds of signs in Semiotics:

1. Icons – simply the sign physically resembles what it stands for.
   Examples: a picture of a person stands for a particular person, a picture of a dog stands for a dog, a “no-smoking” icon sign stands for “no-smoking”, etc. In

this kind of sign, the icon means what it is: you see a person sign – it means the person; you see the dog sign – it means the dog; and you see the “no-smoking” sign – it means “no-smoking”.

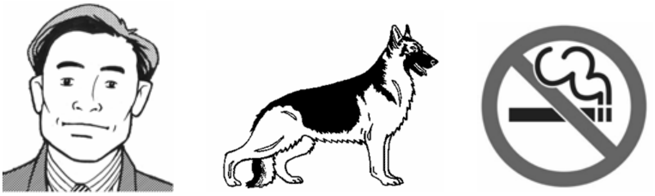


Fig. 5.3 A sign represented by Icons

2. Indexes – indexical signs that are indicators of some fact or condition.

   Examples: a person smiling indicates he is happy, a dog baking indicates it is angry. Different from the Icon sign, in that you did not see “happy” or “angry” from the sign, but you indicate it.

3. Symbols – the sign represents something in a completely arbitrary relationship, and the relationship between the symbol and meaning are subjectively defined. A symbol related to a meaning is just by what it had been defined, but had no any physical resembling meaning likes Icon sign, and also had no logical indication meaning like Index sign. Languages are the most important symbolic sign system in semiotics. In written linguistic system, word is an example of symbols.

   Examples: the words “Human”, “人”, “ひと”, “Dog”, “狗”, “いぬ” stand for what they had been defined by social convention (different forms of word in US, Chinese, and Japanese society). They had neither physical iconic meaning nor indexical meaning, but may represent for the same thing by different symbolic signs.

Linguistic Sign in Saussurean Semiology

Saussure (1983) introduced semiology, and it was focused on linguistic sign, such as word. Semiotics by Peirce (1931-58) is a broader study of language and logic in the branches of syntax, semantics and pragmatics, while saussurean semiology focuses on studying the linguistic sign system, which is important field to analyze the meaning of language system and the creation of ontology by language.

In saussurean semiology, a sign is composed of a “*signifier*” and a “*signified*”. The “*signifier”* can be described as the form that the sign takes (word as a linguistic sign), and the “*signified*” can be described as the concept to which the sign refers. “The linguistic sign does not unite a thing and a name, but a concept and a sound image” (Saussure 1983). The Saussure’s sign model is being composed of:

- A “*signifier*” – the form that the sign takes
- A “*signified*” – the concept to which the sign refers

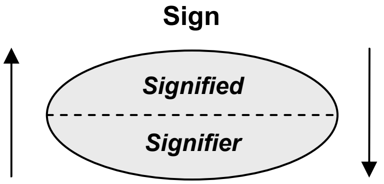


Fig. 5.4 Saussure's sign model

The model of the saussure's sign is shown in Figure 5.4. In the model, the whole sign is the association of the "*signifier*" and the "*signified*". The model of the association between the "*signifier*" and the "*signified*" is defined as "*signification*". The "*signification*" is shown by the arrows from both directions in Figure 5.4.

Example 5.1

- The *signifier* – the word "*Tree*"
- The *signified* concept – the tree (a plant) in the real world

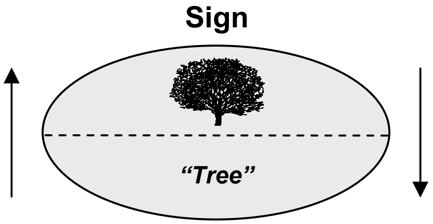


Fig. 5.5 Signification between concept and word

Example 5.2

- The *signifier* – the word "樹"
- The *signified* concept – the tree (a plant) in the real world

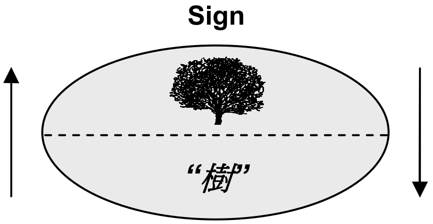


Fig. 5.6 Signification between concept and word

The Sign Relations

An individual sign is not able to reflect things in the real world, but it requires the entire system of sign. (Jameson 1972). The entire system of sign is actually composed of relations between different signs, as a sign has no absolute values, relations between signs can create more value and meaning, as shown in Figure 5.7.

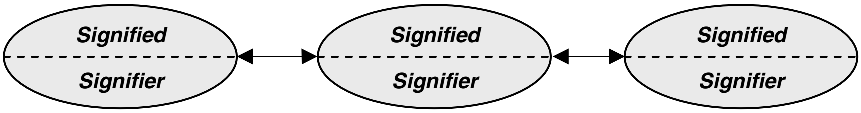


Fig. 5.7 Relations between signs

5.2.2 The Sign System for Concept Formation

A semiotic consists of the relation between a *representamen* (an icon, index, or symbol) and a *referent,* the object to which the *representamen* refers. We create a *interpretant* through this semiotic relation. This relation forms an excitation called *Concept* that is able to identify the symbol as referring to the *referent.* The triadic relation between the *representamen, referent* and *interpretant* refers the meaning triangle (Ogden and Richards 1923) as shown in Figure 5.8.

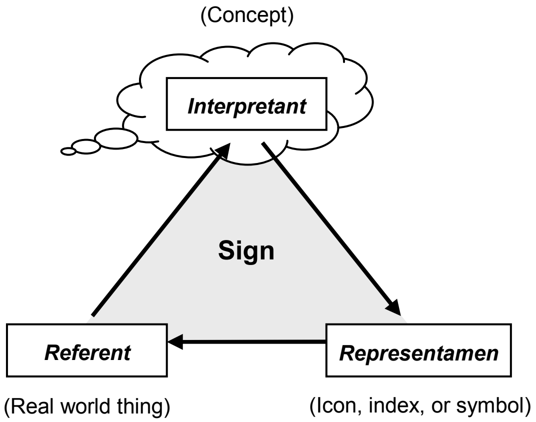


Fig. 5.8 The Meaning Triangle (Ogden and Richards 1923)

The meaning triangle consists of the following to compose a sign:

- The representamen – the kind of the sign, either an icon, an index, or a symbol. In particular, a word can be defined as a symbolic sign for the represenamen.
- The referent – a thing, an object, or an event in the real world that the representamen refers to.
- The interpretant – a concept, or a sense that made by the association of representamen and referent.

Example 5.3

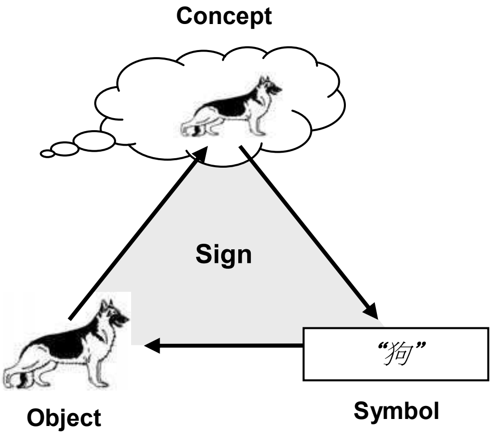


Fig. 5.9 The meaning triangle of the symbol "Dog"

Relations and Semiotics

Integrating the theory of Peirce's semiotics and Sassure's semiology, the sign meaning is formed by relating between different signs and there are three different forms of relation in a sign system (Krieg 2007), as shown in Figure 5.10:

- Object relation (O) – the referent that the sign refers to.
- Representamen relation (R) – the kinds of sign (a word symbol) or the signifier that the sign is represented.
- Interpretant relation (I) – the concept, meaning, or the signified concept that the sign had made by the relation.

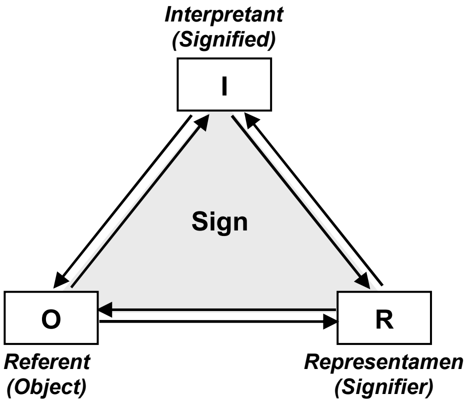


Fig. 5.10 Signs and relations

5.2.3 The Meaning of Words

There are four basic grammatical categories of words:

1. Nouns – linguistic variables that identify general classes of objects, entities, or activities.
   Example: dog, people, house, table, chair, building, age, height…
2. Adjectives – linguistic terms that modify and describe subclasses of entitles.
   Example: big dog, happy people, small house…
3. Adverbs – linguistic terms that qualify subclasses of adjectives
   Example: very big dog, very happy people, really small house…
4. Verbs – grammatical center of predicates that express an act, an event, an occurrence, a mode of being, etc.
   Example: eat, fight, go, make, produce, damage, walk, fly…

Verb is the major part of speech in a sentence. A verb is "a sign of something said of something else, that is, of something either predicable of or present in some other thing" (Aristotle). Verbs are always identified as predicates (formal logic) in a sentence, to indicate a state of being (expressing existence), and action verbs (expressing actions process, events or occurrences) (Abdoullaev 2008). Verbs are there divided into four basic categories:

1. Universal verb – expressing existence
2. State verb – expressing state
3. Action predicates – expressing change or action
4. Relative predicates – expressing relation or association

Ontological and Word Triples

In computing ontology, such as RDF and OWL, concepts and their relationships are defined as ontological triples (also called an ontological statement). The ontological triples are defined as

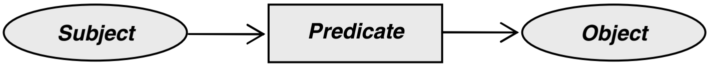


Fig. 5.11 Ontological triples

In the sentence level, the fundamental structure must consist of a noun (or noun phrase) as a subject, a verb as a predicate and another noun (or noun phrase) as an object to commit and express the ontological triples. This gives the fundamental meaning of a sentence in the language system. The verb as a predicate is used to connect and relate all nouns together to express associative meaning. The components in a sentence that make up the triple are defined as:

Sentence → Noun (Subject) + Verb (Predicate) + Noun (Object)

Relations using verb to connect and associate two Nouns can be transformed to the Verb function and the word network for noun is created by more than one triples as shown in Figure 5.12.

Sentence → Verb (Noun, Noun), S → V (N, N)

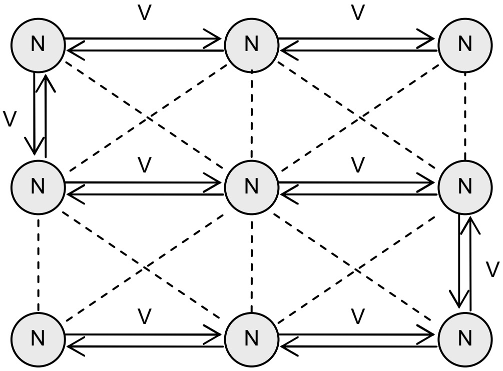


Fig. 5.12 Word network by word triples

Example 5.4

- Sentence = "*Tom has dog*"
- Subject (Noun) = "*Tom*"
- Predicate (Verb) = "*has*"
- Object (Noun) = "*dog*"
- Sentence function = *has (tom, dog)*
- Ontological triple:

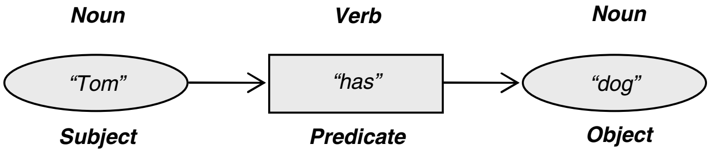


Fig. 5.13 Sentence triple for word network

5.2.4 The Semantics of Relations

Relations are the most essential elements to create concepts. Creating meaning by language also requires analyzing the relations between words in a sentence (or in a text). Words occurring in a single sentence create relationship or association between them (e.g. by sentence triples). Relating those words together is the major method to create concept and meaning. Different types of relations between words create different semantics (meaning). Dictionaries such as WordNet (Miller 1998) and HowNet (Dong 1998) have defined different types of relations for creating associative structure between words (see Chapter 1), for example: the super-ordinate and sub-ordinate (or the hyponymy and hyperonymy) relations are the major relations defined to characterize the meaning of noun in those dictionaries. Recall the Kant's categories presented in Chapter 1, the categories define the concept of understanding as *quantity*, *quality*, *relation*, and *modality*, and it further divides the *relation* in the three sub-categories:

- Inherence and Subsistence (categorical) – the predicate to the subject
- Causality and Dependence (hypothetical) – the cause to its effect
- Community (disjunctive) – the parts of the whole, or the relation of community

Coherence Relations

Coherence cannot exclusively lie in the text (Gernsbacher 1990), and the coherence relations are used to represent the construction of a coherent mental of the situations described by the text (Louwerse 2002). Coherence relations are expressed in three different types:

Types

- Causal – the cause-effect relations of two events provide the basis for rational decision making in human thinking. Cause-effect relations in text are mostly implicit but there are some linguistic expression in text explicitly expressing this type of relations:

  Examples: *A so B, A because B, A therefore B, A since B, A hence B…*

- Temporal – the relation involving time. Knowledge about the temporal order in text importantly expressing how two events relate to each other.

  Examples: *A before B, A after B, A while B, A when B, A until B…*

- Additive – the relation between two events is based on their equivalency, either conjunctive or comparative. In other words, the addictive relation can be used to express the relevancy between two events or objects with respect to others as a whole.

  Examples: *A further B, A moreover B, A similarly B, A alternatively B…*

Polarities and Directions

Two different kinds of polarities

- Positive – *A*⟶ *B (+)*
- Negative – *A*⟶ *B (–)*

Three different kinds of directions

- Forward – *A* → *B*
- Backward – *A* ← *B*
- Bi-directional – *A* ↔ *B*

Table 5.1 Coherence Relations (Louwerse 2001, Mancini and Shum 2006)

Type	Polarity	Direction	Examples
Casual	Positive	Backward	*A because B*
		Forward	*A so B*
		Bi-directional	/
	Negative	Backward	*A although B*
		Forward	*A nevertheless B*
		Bi-directional	/
Temporal	Positive	Backward	*A before B*
		Forward	*A after B*
		Bi-directional	*A while B*
	Negative	Backward	*A until B*
		Forward	*Until A. B*
		Bi-directional	/
Addictive	Positive	Backward	/
		Forward	*A moreover B*
		Bi-directional	*A similarly B*
	Negative	Backward	/
		Forward	*A however B*
		Bi-directional	*A alternatively B*

5.3 Ontology Graph Overviews

The Ontology Graph is a novel approach used in KnowledgeSeeker system to model the ontology of knowledge in text or in an application domain. The Ontology Graph consists of different levels of conceptual units, in which they are associated together by different kinds of relations. It is basically a lexicon system (terms) that linked up among each other to represent a group (a cluster), to formulate concepts and represent meanings. The conceptual structure of an Ontology Graph consists of many terms with some relationships between them, so that different conceptual units are formed like a network model, as shown in Figure 5.14:

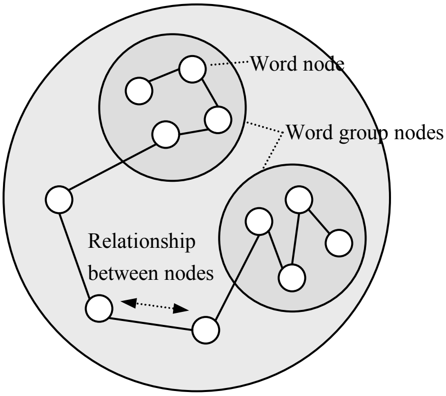


Fig. 5.14 Conceptual units as a network model

The Ontology Graph model mainly consists of two types of objects, they are: 1. Nodes – representing terms, and 2. Relations – representing associations between nodes. These two components in the Ontology Graph define the basics of conceptualizing knowledge in a computer processable form.

5.3.1 Nodes in Ontology Graph

Nodes in Ontology Graph are defined in two different types:

- *Term Node* – An individual term node in the Ontology Graph. It is the most basic conceptual unit in the Ontology Graph represented by a single term, a meaningful term (a sequence of characters), which contribute to defined concepts.
- *Concept Node* – Multiple term nodes grouped in a cluster in the Ontology Graph. The concept node is formulated by any cluster of nodes with relations, representing a certain concept by grouping some high semantically similar terms together.

Example 5.5

Figure 5.15 shows an example of a group of nodes with the following nodes of terms and relations:

- A set of nodes $N = \{n_1, n_2, n_3, n_4\}$ represents four terms where n_1="*David*", n_2= "*Eat*", n_3= "*Apple*", and n_4= "*Happy*".
- A set of relations $R=\{r_1, r_2, r_3\}$ represent the links between nodes where $r_1 = n_1 \times n_2$, $r_2 = n_2 \times n_3$, $r_3 = n_1 \times n_4$

By the definition of concept formation, any group of nodes with relation in the example includes: (n_1, n_2), (n_1, n_2), (n_1, n_2, n_3), (n_1, n_2, n_3, n_4), etc… and therefore we can define concept nodes for $M = \{m_1, m_2, m_3, m_4, ,...\}$ from the example. Four examples of formulated concept nodes are illustrated as following:

- Concept $m_1 = (N_1, R_1)$ where $N_1 = \{n_1, n_4\}$ and $R_1 = \{r_3\}$ representing:
  n_1:"*David*" → n_4:"*Happy*" or can be written as: "*David is happy*".

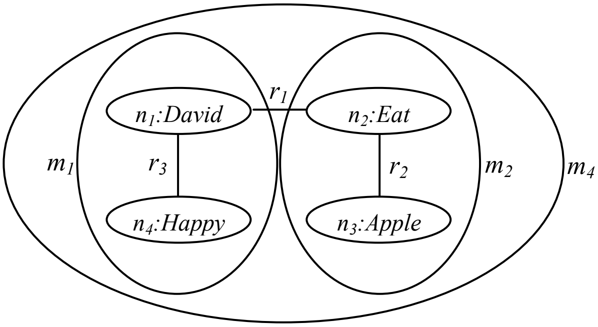


Fig. 5.15 A formation of meaning with terms - example

- Concept $m_2 = (N_2, R_2)$ where $N_1 = \{n_2, n_3\}$ and $R_1 = \{r_2\}$ representing: n_2: "*Eat*" → n_3:"*Apple*" " or can be written as: "*Apple is eaten*".
- Concept $m_3 = (N_3, R_3)$ where $N_1 = \{n_1, n_2, n_3\}$ and $R_1 = \{r_1, r_2\}$ representing: n_1:"*David*" → n_2:"*Eat*" → n_3:"*Apple*" " or can be written as: "*David eats the apple*".
- Concept $m_4 = (N_4, R_4)$ where $N_1 = \{n_1, n_2, n_3, n_4\}$ and $R_1 = \{ r_1, r_2, r_3\}$ representing: n_4:"*Happy*" → n_1:"*David*" → n_2: "*Eat*" → n_3:"*Apple*" " or can be written as: "*David eat the apple happily*".

Besides defining the nodes as a word sign, a node can be also defined by a symbol (a symbol can be regarded as a picture also). As shown in Figure 5.16, the word sign - "*Apple*" can be replaced by a symbol sign, which actually has the same meaning to the word "*Apple*", and the word sign - "*David*" can be also replaced by the real photo, which actually representing the same meaning of the same person named "*David"*. In this situation, everything in the world is just a sign, different sign can be signified to a concept, no matter the sign is in what form (it can be a word, a symbol, a photo of a person, or whatever it can signify concepts). Therefore, both Figures 5.15 and 5.16 actually contribute to the same meaning as a whole, with four same concepts comprised of:

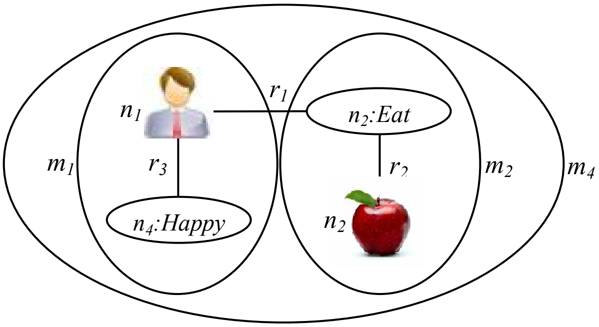


Fig. 5.16 A formation of meaning with symbols - example

5.3.2 Term Nodes in Ontology Graph

To define a lexical word as a term node in Ontology Graph, we need to select a word that is "meaningful" in human perspective. In natural language system, the four basic grammatical categories of words are: noun, adjective, adverb, and verb. However, we select only three of them to be included as term node in creating Ontology Graph. They are: noun, adjective and verb. Nouns are also divided into common noun and proper noun according to its nature. Other words including adverb are filtered and excluded from the Ontology Graph, as shown in the following:

Inclusion of Term Node in the Ontology Graph

Words that are valid to represent a *Term Node* are normally defined by the following part-of-speeches (POS):

- *Common Noun* - A term that refers and describes a person, place, thing, state or quality, etc.
  Examples: dog, people, house… / "My *dog* in the *house*"
- *Proper Noun* – A term that name people, places, and things.
  Example: David, Polytechnic University, Sony… / "*David* eats the apple"
- *Adjective* - A descriptive terms that describe and modify the meaning of a noun
  Example: big, happy, fast… / "David is *happy*"
- *Verb* – A term that describes an action or a state
  Example: eat, fight, go…/ "David *eats* the apple"

Exclusion of Term Node in the Ontology Graph

Non-meaningful words that are filtered and excluded in representing a *Simple Node* are defined by the other part-of-speeches (POS) as follow:

- *Adverb* – A term that describe a verb, adjective or adverb
  Examples: very, really, happily… / "David eats the apple *happily*"
- *Pronoun* – Replace a noun
  Examples: she, he, they… / "*My* dog is in the house"
- *Preposition* – Links a noun to other words
  Examples: to, in, for… / "My dog is *in* the house"
- *Conjunction* – Joins two words, clauses or sentences
  Examples: and, but, so… / "David eats the apple *and* David is happy"
- *Interjection* – A short exclamation in a sentence
  Examples: well, hi, oops… "*Hi*! How are you?"

These types of word which are excluded are normally defined as stop-word that are removed from the information retrieval processing, i.e. it is assumed there is no (or less) contribution to defining meaning for including these words for the information retrieval system. The POS of words in Ontology Graph are summarized in Table 5.2:

Table 5.2 Summary of POS in IG

POS	Examples	Inclusion
Common Noun	dog, people, house	+
Proper Noun	David, Polytechnic University, Sony	+
Adjective	big, happy, fast	+
Verb	eat, fight, go	+
Adverb	very, really, happily	–
Pronoun	she, he, they	–
Preposition	to, on, for	–
Conjunction	and, but, so	–
Interjection	well, hi, oops	–

Therefore, Ontology Graph contains the following words for representing a Term Node:

- N_{CN} – A node which is represented by a common noun of word
- N_{PN} – A node which is represented by a proper noun of word
- N_{ADJ} – A node which is represented by an adjective of word
- N_V – A node which is represented by a verb of word

The Example 5.5 is thus modified as follows (Example 5.6):

Example 5.6

- A set of nodes $N = \{n_{1,} n_{2,} n_{3,} n_4\}$ represents the words where n_1=“*David*”/$N_{PN,}$ n_2= “*Eat*”/$N_{V,}$ n_3= “*Apple*”/$N_{CN,}$ and n_4= “*Happy*”/N_{ADJ}.
- A set of relations $R=\{r_1, r_2, r_3\}$ represents the links between nodes where $r_1 = n_1 \times n_2$, $r_2 = n_2 \times n_3$, $r_3 = n_1 \times n_4$

Four different types of relations are represented in Table 5.3:

Table 5.3 Types of relations - example

Relations	Details	Descriptions
r_1	$N_{PN} \times N_V$	Proper Noun to Verb
r_2	$N_V \times N_{CN}$	Common Noun to Verb
r_3	$N_{PN} \times N_{ADJ}$	Proper Noun to Adjective

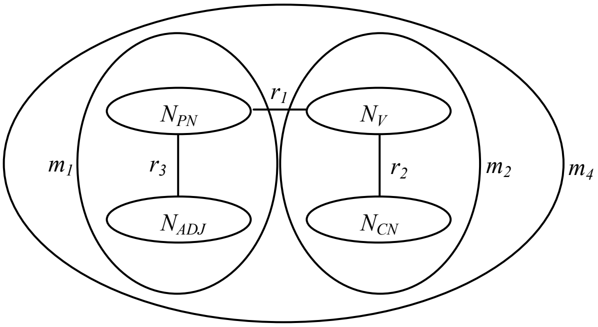


Fig. 5.17 Example of meaning formation

5.3.3 Words Function

Differentiating every word node from the kinds of $\{N_{CN}, N_{PN}, N_{ADJ}, N_V\}$ aims to model the different functions of language. Different POS of word plays different role in language for communication. Although the use of language in text is very vague, and different POS of word does not guarantee to express a particular language function explicitly, the POS of word is still playing an important role to reflect different kinds of meaning, especially for the two types of function – understanding and feeling.

5.3.3.1 The Function of Language

The function of language established by Jakobson R consists of six elements as summarized in Figure 5.18 and Table 5.4:

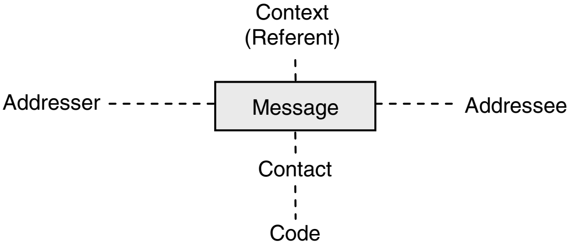


Fig. 5.18 Jakobson's model of the function of language

Table 5.4 Factors of Communication and Functions of Language

No.	Factor	Function	Purpose
1	Context	Referential	Expressing information
2	Addresser	Emotive	Expressing feelings or emotions
3	Addressee	Cognitive	Expressing influence
4	Code	Metalingual	Expressing interaction
5	Contact	Phatic	Establishing social relationship
6	Message	Poetic	Part of the message

5.3.3.2 Understanding and Feeling

Among the six factors of communication as shown in Table 5.4, our approach focuses on the context and addresser factor, which correspondingly refer to the referential and emotive function. Analyzing these two functions of language are useful for extracting information and emotion expression in text, and they are used to model the knowledge of understanding and feeling. The simplified knowledge definition by referential function and emotive function are described as follows:

- Referential function – a function describing objective or cognitive of the world
- Emotive function – a function describing subjective and expressive of a person

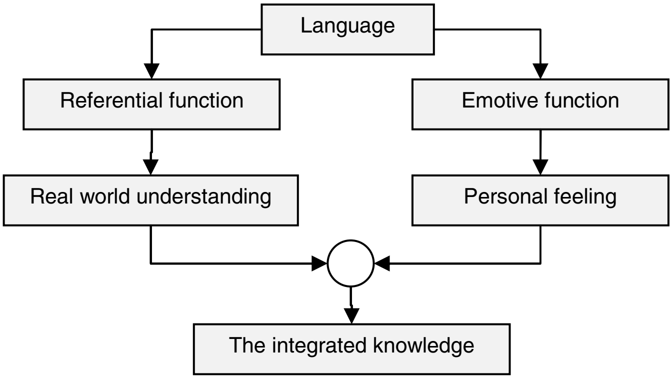


Fig. 5.19 The functions of language and knowledge

The objective expression (referential function) in language means to describe the general understanding about things in the real world, such as facts, objects, or events, etc. It is relevant to express the knowledge about an objective domain (areas of arts, science, history, etc). The subjective expression (emotive function) in language means to describe the personal feelings of people, such as behavior, emotion, or passion, etc. It is relevant to express the knowledge about a person, every person may have their unique personal knowledge besides the knowledge of some object domain, and there are differences between each other. In other words, the referential function can be used to analyze and define the concept of domain ontology which is more "objective", while the emotive function can be used to analyze and model the concept of personal ontology (about the feeling of a person) which is comparatively more "subjective".

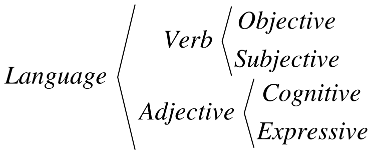


Fig. 5.20 The functions of language and knowledge

Noun is a self-described symbol about an object in the real word, Verb and adjective in language can be used as predicates or functions of noun, e.g. *Sentence → Verb (Noun, Noun)* and *Sentence → Adjective (Noun)*. Both verb and adjective can be classified into two types of function that are referential function and emotive function (Table 5.5). Referential function of verb is used to express objective knowledge; emotive function of verb is used to express subjective knowledge. Referential function of adjective is used to express cognitive knowledge; emotive function of adjective is used to express expressive knowledge. The classification is expressed in Figure 5.20, and the examples are given in Table 5.6. The word-link is denoted by *A—ref→ B* for referential link, and *A—emo→* B for emotive link, and are transformed to the following functions:

- *REF_VERB(A, B)* – Objective expression
- *EMO_VERB(A, B)* – Subjective expression
- *REF_ADJ(A, B)* – Cognitive expression
- *EMO_ADJ(A, B)* – Expressive expression

Table 5.5 Referential and Emotive function of Verb and Adjective

	Verb	Adjective
Referential function	OBJECTIVE	COGNITIVE
Emotive function	SUBJECTIVE	EXPRESSIVE

Table 5.6 Factors of Communication and Functions of Language

Type	Function	Expression	Example
Verb	Referential function	Objective	eat, play, see
	Emotive function	Subjective	love, hate, surprise, fear
Adjective	Referential function	Cognitive	fast, tall, heavy, green
	Emotive function	Expressive	good, beautiful, interesting, cute

5.3.3.3 Meaning and Information

According to the word-link stand in the logical of inclusion, intersection, or exclusion, they are classified into three types of association functions (Figure 5.21: 1. Taxonomic, 2. Semantic, and 3. Diacritical (Guiraud 1971).

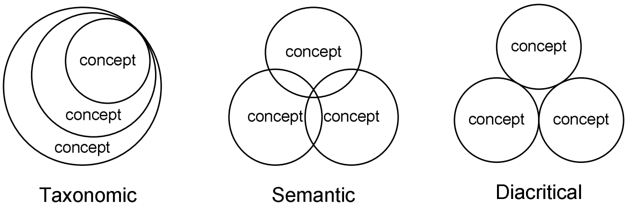


Fig. 5.21 Relation types expressing different nature of knowledge

Every type of words and word-links are therefore further classified into the dimension of associative functions, and different grammatical categories of words are limited to be associated with different association functions:

- Taxonomic function (Inclusion) – only the same typed grammatical categories of words can be associated together.
  Examples: *Mammal (NOUN) → Vertebrate (NOUN), Good (EMO_ADJ) → Beautiful (EMO_ADJ), Green (REF_ADJ) → Light Green (REF_ADJ), Color (NOUN) → Green (NOUN)*

- Semantic function (Intersection) – any grammatical categories of words can be associated together.
  Examples: *Tree (NOUN) → Leave (NOUN), Leave (NOUN) → Green (REF_ADJ), Beautiful (EMO_ADJ) → Tree (NOUN), Tree (NOUN) → See (REF_VERB)*

- Diacritical function (Exclusion) – according to the word association in taxonomic and semantic functions
  Examples: *Tree (NOUN) → Leg (NOUN), Bad (EMO_ADJ) → Beautiful (EMO_ADJ), Leave (NOUN) → Gold (REF_ADJ), Tree (NOUN) → Eat (REF_VERB)*

5.4 The Implementation of Ontology Graph

The actual implementation of an Ontology Graph (OG) adopts the theory and definitions of above discussed Ontology Graph model. The conceptual representation and the class implement hierarchy are given in this section. The implementation of Ontology Graph is used as the fundamental knowledge representation model in KnowledgeSeeker, for ontology storage, learning, querying, and building ontology-based applications.

5.4.1 The Conceptual Structure of Ontology Graph

Figure 5.22 presents the conceptual view of Ontology Graph which is created based on the structure of term nodes and relations. The Ontology Graph consists of four types of *Conceptual Units (CU)* according to their level of complexity exhibiting in knowledge. We define four *Conceptual Units (CU)* – any objects (nodes) in the Ontology Graph that give semantics expression. All of these Conceptual Units are linked up and associated by *Conceptual Relation (CR)* within each other, to comprise the entire conceptual structure of Ontology Graph, and to model an area (a domain) of knowledge.

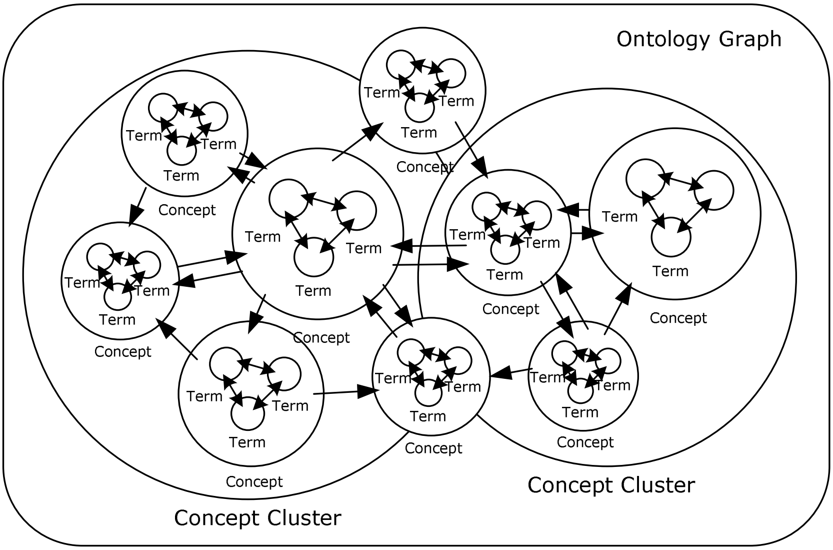


Fig. 5.22 Conceptual structure of Ontology Graph (OG) in KnowledgeSeeker

5.4.1.1 Conceptual Units in Ontology Graph

The four *Conceptual Units (CU)* definitions, their natures and the levels of knowledge according to their complexity are described as follows:

- *Term (T)*. The smallest conceptual unit that extracted in the form of a meaningful word (a sequence of characters), those consist of "meaning" in human perspective.
- *Concept (C).* A number of *Term (T)* grouped together with *Conceptual Relations (CR)* between each other form a *Concept (C)*, it is the basic conceptual unit in the *Concept Graph (CG).*
- *Concept Cluster (CC)*. A number of *Concept (C)* related to each other form a *Concept Cluster (CC)*. It groups similar meaning of concepts in a tight cluster representing a hierarchy of knowledge.
- *Ontology Graph (OG)*. The largest, entire conceptual unit grouped by *Concept Clusters (CC)* is defined as *Ontology Graph (OG)*. It represents a comprehensive knowledge of a certain domain.

5.4.2 The Class Diagram of Ontology Graph

The implementation of Ontology Graph can be represented by a class relationship structure. Different levels of conceptual unit in Ontology Graph are represented by different classes in the implementation. The class diagram (relationships and hierarchies) is shown in Table 5.7 and Figure 5.23.

Table 5.7 Class Relations of Ontology Graph Implementation

Level	Conceptual Units	Class	Relations
Domain level	Ontology Graph	OntologyGraph	OntologyGraph → InterdependencyGraph → ConceptNode [WordNode / ConceptCluster]
Group level	Concept Cluster	ConceptCluster	ConceptCluster → ComplexNode → Word
Concept level	Concept	ComplexNode	ComplexNode → Word
Lexicon level	Word	SimpleNode	SimpleNode → Word [Verb / Adjective / Noun]

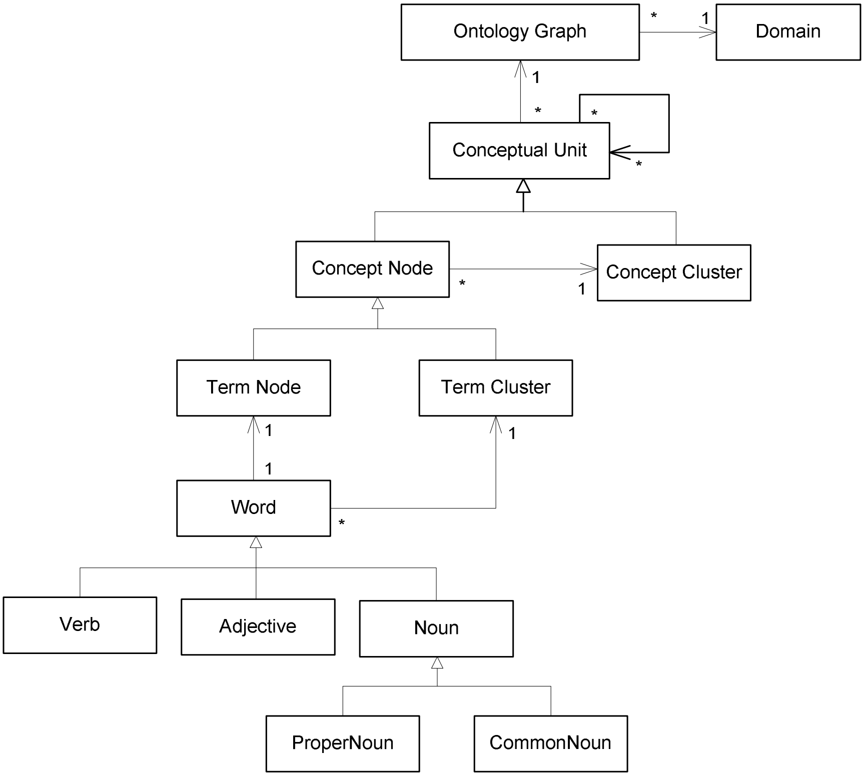


Fig. 5.23 The class relationship of Ontology Graph

Chapter 6
Ontology Learning in Chinese Text

Abstract. In this chapter, an ontology learning process that based on chi-square statistics is proposed for automatic learning an Ontology Graph from texts for different domains. The ontology learning method is illustrated by different steps and examples, and finally we give an experiment which applied the proposed method for automatic learning ten Ontology Graphs to represent ten different domains of knowledge.

6.1 The Ontology Learning Method

The ontology learning is the process to learn and create a domain of knowledge (a particular area of interest such as art, science, entertainment, sport, etc.) in the form of Ontology Graph, which is a knowledge representation model described in the previous chapter. The Ontology Graph creation is considered as a knowledge extraction process. As described in Chapter 3.2, we defined different levels of knowledge objects, in the form of *Conceptual Unit (CU)*, which are required for extraction in the learning process. We define a bottom-up ontology learning approach to extract *Conceptual Units* and create Ontology Graph. The approach identifies and generates *Conceptual Units* from the lowest level, *Term (T)*, to the highest level, the *Ontology Graph (OG)*.

We focused on ontology learning in Chinese Text, because the relationships between Chinese words are more difficult to be analyzed simply by grammar and word pattern (such as by regular expression) than English word. Therefore, we use Chinese text as the knowledge source to learn and create Ontology Graph which can reveal the feasibility and effectiveness of learning ontology based on term relations, through the proposed learning approach.

The five learning sub-processes start from the bottom, are defined as

1. *Term extraction* – the most basic process that recognizes meaningful Chinese terms in text documents.
2. *Term-to-class relationship mapping* – the second process that finds out the relations between terms and classes (domain).
3. *Term-to-term relationship mapping* – the third process that finds out the relations between all Chinese terms within a class (domain).
4. *Concept clustering* – the fourth process which further groups (clusters) the Chinese terms within a class (domain) based on their similarity.

E.H.Y. Lim, J.N.K. Liu, and R.S.T. Lee: Knowledge Seeker, ISRL 8, pp. 71 – 98.
springerlink.com

5. *Ontology Graph generation* – the final process that generates a graph-based Ontology Graph as knowledge representation for application use.

Figure 6.1 shows all the sub-processes in the bottom-up approach of the ontology graph learning method. All of these sub-processes correspond to identifying different levels of *Conceptual Unit (CU).* Thus the knowledge is learnt from the smallest *CU (Term, T)* towards the largest *CU (Ontology Graph, OG).*

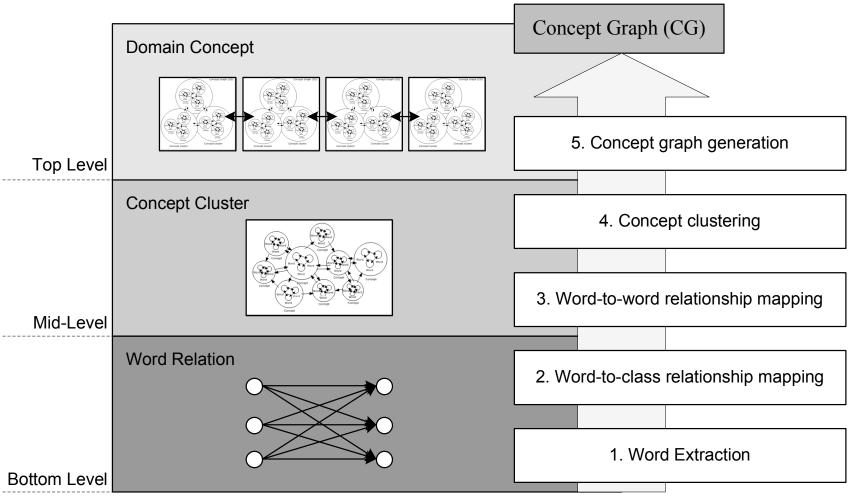


Fig. 6.1 Bottom-up approach of the Ontology Graph learning process

6.1.1 Term Extraction

Our approach focuses on learning Ontology Graph from Chinese text and thus the prepared text corpus entirely consists of Chinese texts. Since Chinese writing does not separate words with a space, a useful means of word disambiguation is not available in Chinese that is available in English. For this reason, Chinese term extraction typically relies on dictionaries. An existing electronic dictionary is available such as HowNet (Dong and Dong 1998). It contains over 50000 distinct Chinese words and it is useful to identify a meaningful word inside a text, and it can serve as our initial term list for doing term extraction process. By applying a maximal matching algorithm to the word list and a set of Chinese text corpus, we can extract a candidate term list (a list of terms that are potentially of a relevant concept and thus to be extracted for the learning process), while filtered out other unnecessary terms that do not appear in the text corpus. N numbers of candidate terms $T = \{t_1 ... t_n\}$ are thus extracted, where every term t_i in the term list T appears at least once in the text corpus.

Besides the existing terms in the dictionary, an additional input of Chinese terms into the term extraction process is also required. These additional words, such as named person/organization, brand/building names, new technologies, usually are not maintained in the dictionary since the dictionary is not keeping updates all the time. Therefore, adding new terms into the initial word list by human effort is required.

6.1.2 Term-to-Class Relationship Mapping

The candidate term list *T* extracted from the Chinese text corpus however has no meaning and relationship to any conceptual units in the Ontology Graph model. So the second process that applied to the candidate term list is the term-to-class relationship mapping. This mapping process acts like feature selection that it selects and separates every term in the term list to its most related domain class. First of all we need to prepare a set of labeled text corpus (a set of text documents which are classified into different labels of class or domain topic). Then we can measure how the terms are related to each class, and select a sub-list of terms in the candidate term list for each class. The mapping process means that we put every term in the sub-list associated with a class, by a weighted and directed relation between a term and a class, as shown in Figure 6.2.

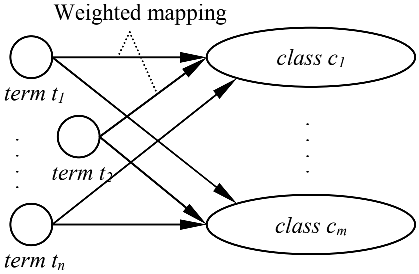


Fig. 6.2 Term mapping to classes

6.1.2.1 Term-to-Class Independency Measurement by χ^2

The term-to-class relationship mapping applies a χ^2 statistical term-to-class independency measurement to measure the degree of interdependency between a term and a class. The measurement is carried out by first calculating the co-occurrence frequencies between every term t and class c. It is expressed in a two-way contingency table as shown in Table 6.1.

Table 6.1 2 x 2 term-to-class contingency table of term t and class c

	c	$\neg c$	Σ
t	$O_{t,c}$	$O_{t,\neg c}$	$O_{t,c} + O_{t,\neg c}$
$\neg t$	$O_{\neg t,c}$	$O_{\neg t,\neg c}$	$O_{\neg t,c} + O_{\neg t,\neg c}$
Σ	$O_{t,c} + O_{\neg t,c}$	$O_{t,\neg c} + O_{\neg t,\neg c}$	$O_{t,c} + O_{t,\neg c} + O_{\neg t,c} + O_{\neg t,\neg c} = N$

The term-to-class contingency table is comprised of the cells of observed frequency $O_{i,j}$ where $i \in \{t, \neg t\}$ and $j \in \{c, \neg c\}$. Thus, $O_{t,c}$ is the observed frequency (number) of documents in class c which contains the term t; $O_{t, \neg c}$ is the observed frequency of documents which are not in class c and contain the term t; $O_{\neg t,c}$ is the observed frequency of documents which are in class c and do not contain the term t; and $O_{\neg t, \neg c}$ is the observed frequency of documents which are neither in class c nor contain the term t.

The observed frequency is compared to the expected frequency $E_{i,j}$ where $i \in \{t, \neg t\}$ and $j \in \{c, \neg c\}$. $E_{i,j}$ is defined as

$$E_{i,j} = \frac{\sum_{a \in \{t, \neg t\}} O_{a,j} \sum_{b \in \{c, \neg c\}} O_{i,b}}{N}$$

χ^2 statistical independency measurement for term t and class c is defined as

$$\chi^2_{t,c} = \sum_{i \in \{t, \neg t\}} \sum_{j \in \{c \neg c\}} \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}}$$

Figure 6.3 summarizes the steps of the term-to-class relationship mapping process. In the first step, a set of labeled document corpus is prepared. The related class labels are identified from the text corpus, and then a candidate term list is extracted from the same text corpus. Every term in the candidate term list has no relationship to any class labels, since one single term may exist in more than one class. Therefore, the process of the term-to-class independency measurement is processed to classify and associate every term in the candidate term list with the most relevant class by the χ^2 statistical measurement.

Steps of Term-to-Class Relationship Mapping Process

```
STEP 1: Prepare a labeled document corpus $D$
                $D = \{d_1, d_2, ......\}$
STEP 2: Extract set of distinct classes $C$ from the corpus
                $C=\{c_1, c_2, ......\}$
STEP 3: Extract candidates term list $T$ from the corpus
                $T=\{t_1, t_2, ......, t_n\}$
STEP 4: Independency measurement for every term to class
        For each class $c$ in $C$
            For each term $t$ in $T$
                Calculate $\chi^2_{t,c}$
            Next
        Next
```

Fig. 6.3 Term-to-class Relationship mapping steps

Example 6.1

Take an example of a χ^2 statistical measurement on 10 documents in 5 classes with 8 candidate terms as described in Table 6.2 and Figure 6.4, the term-to-class dependency matrix is transformed as shown in Table 6.3.

Table 6.2 Content of term and document distribution

Document	Term occurrence	Class
d_1	t_1, t_2	c_1
d_2	t_2	c_1
d_3	t_2, t_3, t_4	c_1
d_4	t_3	c_2
d_5	t_6	c_2
d_6	t_4, t_5, t_6	c_3
d_7	t_1, t_5, t_6	c_3
d_8	t_6	c_4
d_9	t_6, t_7, t_8	c_4
d_{10}	t_5, t_8	c_5

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

Fig. 6.4 8 x 10 term-by-document matrix for 10 documents and 8 terms

Table 6.3 Term-to-class table - 10 documents in 5 classes with 8 candidate terms

	c_1	c_2	c_3	c_4	c_5
t_1	d_1		d_7		
t_2	d_1, d_2, d_3				
t_3	d_3	d_4			
t_4	d_3		d_6		
t_5	d_3		d_6, d_7		d_{10}
t_6		d_5	d_6, d_7	d_8, d_9	
t_7				d_9	
t_8				d_9	d_{10}

- Document corpus (number of documents = 10): $D = \{d_{1,} d_2,\ldots, d_{10}\}$
- Labeled classes (number of classes = 5): $C = \{c_1, c_2,\ldots, c_5\}$
- Extracted candidate terms (number of terms = 8): $T = \{t_1, t_2,\ldots, t_8\}$

A χ^2 mapping table that maps the candidate terms to classes can be formed as shown in Table 6.4. This table expresses the relationship of the term t_i and the class c_i by the χ^2 weight. This weight measures how the term is related to the class. By selecting the highest weight of every term-to-class mapping entry (highlighted in Table 6.4), every candidate term is now mapped to a single class, as shown in Figure 6.5.

Table 6.4 χ^2 mapping of 5 classes and 8 candidate terms

	c_1	c_2	c_3	c_4	c_5
t_1	0.476	0.625	1.406	0.625	0.278
t_2	10.000	1.071	1.071	1.071	0.476
t_3	0.476	1.406	0.625	0.625	0.278
t_4	0.476	0.625	1.406	0.625	0.278
t_5	0.079	1.667	3.750	1.667	1.667
t_6	4.286	0.000	2.500	2.500	1.111
t_7	0.476	0.278	0.278	4.444	0.123
t_8	1.071	0.625	0.625	1.406	4.444

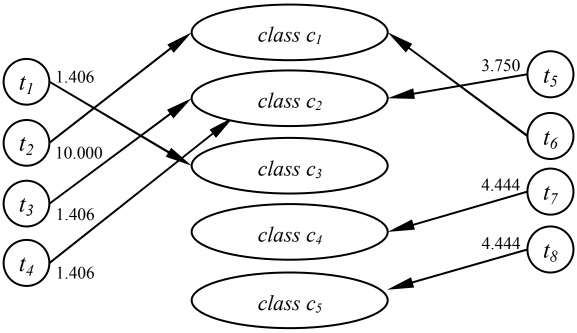


Fig. 6.5 Term mapping to classes from the example

This χ^2 calculation may contain incorrect results and may not fully explore all the valid mapping in the text corpus. For example, the term t_6 has the highest mapping value to c_1 among all the classes (c_1, $c_{2,}$ c_3, c_4, c_5), however the term t_6 actually does not exist once in class c_1. This situation is illustrated in the Example 6.2:

Example 6.2

Table 6.5 and Table 6.6 show the observed frequency of term t_6 to classes c_1 and c_3 from the example given in Table 6.3:

Table 6.5 2x2 term-to-class contingency table of term t_6 and class c_1

	c_1	$\neg c_1$	Σ
t_6	*0*	*5*	*5*
$\neg t_6$	*3*	*2*	*5*
Σ	*3*	*7*	*10*

Table 6.6 2x2 term-to-class contingency table of term t_6 and class c_3

	c_3	$\neg c_3$	Σ
t_6	*2*	*3*	*5*
$\neg t_6$	*0*	*5*	*5*
Σ	*5*	*5*	*10*

Applying the mapping equation to Table 6.5, we produce: $E_{t_6,c_1} = 1.5$, $E_{t_6,\neg c_1} = 3.5$, $E_{\neg t_6,c_1} = 1.5$, $E_{\neg t_6,\neg c_1} = 3.5$, and $\chi^2_{t_6,c_1} = 4.286$. For Table 6.6, we produce: $E_{t_6,c_3} = 1$, $E_{t_6,\neg c_3} = 4$, $E_{\neg t_6,c_3} = 1$, $E_{\neg t_6,\neg c_3} = 4$, and $\chi^2_{t_6,c_3} = 2.500$. In this result, we produced $\chi^2_{t_6,c_1} > \chi^2_{t_6,c_3}$ (4.286 > 2.500), and this means the term t_6 has stronger dependency on class c_1 than c_3. However, t_6 has in fact more occurrence in c_3 than in c_1 (2 times in c_3 while 0 in c_1). A zero occurrence in a class can obtain a high χ^2 statistical value meaning that the statistic does not reflect the real situation. Another example (Example 6.3) is given to further illustrate the problem:

Example 6.3

Table 6.7 and Table 6.8 show the observed frequency of word w_6 to class c_1 from Table 6.3:

Table 6.7 2x2 term-to-class contingency table of term t_6 and class c_1

	c_1	$\neg c_1$	Σ
t_1	*1*	*1*	*2*
$\neg t_1$	*2*	*6*	*8*
Σ	*3*	*7*	*10*

Table 6.8 2x2 term-to-class contingency table of term t_6 and class c_3

	c_1	$\neg c_1$	Σ
t_7	*0*	*1*	*1*
$\neg t_7$	*3*	*6*	*9*
Σ	*3*	*7*	*10*

Applying the mapping equation to Table 6.7, we produce: $E_{t_1,c_1} = 0.6$, $E_{t_1,\neg c_1} = 1.4$, $E_{\neg t_1,c_1} = 2.4$, $E_{\neg t_1,\neg c_1} = 5.6$, and $\chi^2_{t_1,c_1} = 0.476$. For Table 6.8 we produce: $E_{t_7,c_1} = 0.3$, $E_{t_7,\neg c_1} = 0.7$, $E_{\neg t_7,c_1} = 2.7$, $E_{\neg t_7,\neg c_1} = 6.3$, and $\chi^2_{t_6,c_3} = 0.4760$. In this study, we produced the same χ^2 statistical result that $\chi^2_{t_1,c_1} = \chi^2_{t_7,c_1} = 0.476$, and this means both terms t_1 and t_7 have the same dependency on class c_1. However, t_1 and t_7 actually have different occurrence distributions in class c_1. Different distribution of occurrences producing the same χ^2 statistical value reveals that the values do not reflect real situation about the term dependency on a class.

6.1.2.2 Term-to-Class Positive and Negative Dependency Measurement by R

The problem of using χ^2 statistic measurement is that we can measure the term dependency on a class, but cannot measure whether the dependency is positive or negative (Li et al. 2008). In example 3.2, although the result showed that $\chi^2_{t_6,c_1} > \chi^2_{t_6,c_3}$ (4.286 > 2.500) for the word t_6, there is 0 occurrence among all documents in class c_1 (d_1, d_2, d_3) (0 out of 3 = 0%), and also there is 0 document that containing t_6 has been classified as class c_1 (0 out of 5 = 0%). Therefore, we define this dependency as negative dependency, i.e. term t_6 has a negative dependency on class c_1. On the other hand, the term t_6 has an occurrence of 2 among all 2 documents in class c_3 (d_6, d_7) (2 out of 2 = 100%), and also there are 2 documents that containing t_6 have been classified as class c_3 (2 out of 3 = 66%). Therefore, we define this dependency as positive dependency, i.e. term t_6 has a positive dependency on class c_3. Similarly in example 3.3, although the result showing that $\chi^2_{t_1,c_1} = \chi^2_{t_7,c_1} = 0.476$, term t_7 actually has a negative dependency on the class c_1 and term t_1 has a positive dependency on class c_1. The measurement of a term dependency on a class is whether negative or positive, is defined by the equation of ratio between observed frequency and expected frequency, as $R_{t,c}$ (Li et al. 2008):

$$R_{t,c} = \frac{O_{t,c}}{E_{t,c}}$$

$R_{t,c}$ can be defined as:

$$R_{t,c} = \frac{p(t,c)p(\neg t,\neg c) - p(t,\neg c)p(\neg t,c)}{p(t)p(c)} + 1$$

$R_{t,c}$ is the ratio between $O_{t,c}$ and $E_{t,c}$. Term t is measured as positive dependency on class c if $R_{t,c} > 1$, or term t is measured as negative dependency on class c if $R_{t,c} < 1$. $R_{t,c} = 1$ means that there is no dependency between t and c. In summary,

$\chi^2_{t,c}$ measures the dependency between a term and a class in a corpus distribution, while $R_{t,c}$ measures whether the dependency is positive or negative:

$$t_i \begin{cases} \textit{negative dependency to } c_j \textit{ if } R_{t,c} < 1 \\ \textit{positive dependency to } c_j \textit{ if } R_{t,c} > 1 \end{cases}$$

Figure 6.6 presents the updated steps of the term-to-class mapping process:

```
Updated Steps of Term-to-Class Relationship Mapping Process
STEP 1: Prepare a labeled document corpus $D$
STEP 2: Extract set of distinct classes $C$ from the corpus
STEP 3: Extract candidate term list $T$ from the corpus
STEP 4: Independency measurement for every term to class
STEP 5: Positive/negative measurement for every term
          For each class $c$ in $C$
              For each term $t$ in $T$
                  Calculate $R_{t,c}$
              Next
          Next
```

Fig. 6.6 Updated term-to-class relationship mapping steps

Example 6.4

To determine whether the term has negative or positive dependency on a class, the Example 6.1 is extended by further measuring the $R_{t,c}$ values, the result is shown in Table 6.9. For every term in $T = \{t_1, t_2,\ldots, t_8\}$ to class $C = \{c_1, c_2,\ldots, c_5\}$ the dependency value of the example is calculated and summarized in Tables 6.10 to 6.17 correspondingly to the terms t_1 to t_8.

Table 6.9 $R_{t,c}$ mapping of 5 class and 8 candidate terms

	c_1	c_2	c_3	c_4	c_5
t_1	1.667	0.000	2.500	0.000	0.000
t_2	3.333	0.000	0.000	0.000	0.000
t_3	1.667	2.500	0.000	0.000	0.000
t_4	1.667	0.000	2.500	0.000	0.000
t_5	0.833	0.000	2.500	0.000	2.500
t_6	0.000	1.000	2.000	2.000	0.000
t_7	0.000	0.000	0.000	5.000	0.000
t_8	0.000	0.000	0.000	2.500	5.000

Table 6.10 Dependency values of term t_1

class	$\chi^2_{t,c}$	$R_{t,c}$	*dependency*
c_1	0.476	1.667	positive
c_2	0.625	0	negative
c_3	1.406	2.500	positive
c_4	0.625	0	negative
c_5	0.278	0	negative

Table 6.11 Dependency values of term t_2

class	$\chi^2_{t,c}$	$R_{t,c}$	*dependency*
c_1	10	3.333	positive
c_2	1.071	0	negative
c_3	1.071	0	negative
c_4	1.071	0	negative
c_5	0.476	0	negative

Table 6.12 Dependency values of term t_3

class	$\chi^2_{t,c}$	$R_{t,c}$	*dependency*
c_1	0.476	1.667	positive
c_2	1.406	2.500	positive
c_3	0.625	0	negative
c_4	0.625	0	negative
c_5	0.278	0	negative

Table 6.13 Dependency values of term t_4

class	$\chi^2_{t,c}$	$R_{t,c}$	*dependency*
c_1	0.476	1.667	positive
c_2	0.625	0	negative
c_3	1.406	2.500	positive
c_4	0.625	0	negative
c_5	0.278	0	negative

Table 6.14 Dependency values of term t_5

class	$\chi^2_{t,c}$	$R_{t,c}$	*dependency*
c_1	0.079	0.833	negative
c_2	1.667	0	negative
c_3	3.750	2.500	positive
c_4	1.667	0	negative
c_5	1.667	2.500	positive

Table 6.15 Dependency values of term t_6

class	$\chi^2_{t,c}$	$R_{t,c}$	*dependency*
c_1	4.286	0	negative
c_2	0	1.000	negative
c_3	2.500	2.000	positive
c_4	2.500	2.000	positive
c_5	1.111	0	negative

Table 6.16 Dependency values of term t_7

class	$\chi^2_{t,c}$	$R_{t,c}$	*dependency*
c_1	0.476	0	negative
c_2	0.278	0	negative
c_3	0.278	0	negative
c_4	4.444	5.000	positive
c_5	0.123	0	negative

Table 6.17 Dependency values of term t_8

class	$\chi^2_{t,c}$	$R_{t,c}$	*dependency*
c_1	1.071	0	negative
c_2	0.625	0	negative
c_3	0.625	0	negative
c_4	1.406	2.500	positive
c_5	4.444	5	positive

Both $\chi^2_{t,c}$ and $R_{t,c}$ values are calculated for each term-class combination, and they are used for the term-to-class relationship mapping process. When there are *n* terms in the candidate term list $T = \{t_1 ... t_n\}$, and *m* topic classes in $C = \{c_1 ... c_m\}$, the number of the calculations of $\chi^2_{t,c}$ and $R_{t,c}$ are equal to *n* * *m* (*m* vector with *n* values in each vector).

The goal of the term-to-class relationship mapping process is to classify every candidate term t_i, where $t_i \in T$, into its most related class c_j, *where* $c_j \in C$. There are *m* term-dependency vectors *V* if document set *D* contains *m* topic classes, $V = \{v_1 ... v_m\}$ for every topic class c_j with $v_j = \{(t_1, \chi^2_{t_1,c_j}, R_{t_1,c_j}), (t_2, \chi^2_{t_2,c_j}, R_{t_2,c_j}), ..., (t_n, \chi^2_{t_n,c_j}, R_{t_n,c_j})\}$:

- Document corpus (number of documents = *k*): $D = \{d_1, d_2, ..., d_k\}$
- Labeled classes (number of classes = *m*): $C = \{c_1, c_2, ..., c_m\}$
- Extracted candidate terms (number of terms = *n*): $T = \{t_1, t_2, ..., t_n\}$
- Term-dependency vectors: $V = \{v_1, v_2, ..., v_m\}$
  for each $v_j = \{(t_1, \chi^2_{t_1,c_j}, R_{t_1,c_j}), (t_2, \chi^2_{t_2,c_j}, R_{t_2,c_j}), ..., (t_n, \chi^2_{t_n,c_j}, R_{t_n,c_j})\}$

The weight of every term t_i in term-dependency vector v_j for class c_j is ranked by $\chi^2_{t_i,c_j}$, and every v_j contains *n* entries. Every v_j is a vector of term-dependency relationship for a particular class.

Example 6.5

From the result of the previous examples (Examples 6.1 – 6.4), the ranked terms in the term-dependency vector of each class are therefore created:

- Term-dependency of class c_1: $v_{c1} = \{(w_2, 10.000, 3.333), (w_1, 0.476, 1.667), (w_3, 0.476, 1.667), (w_4, 0.476, 1.667)\}$
- Term-dependency of class c_2: $v_{c2} = \{(w_3, 1.406, 2.500)\}$
- Term-dependency of class c_3: $v_{c3} = \{(w_5, 3.750, 2.500), (w_6, 2.500, 2.000), (w_1, 1.406, 2.500), (w_4, 1.406, 2.500)\}$
- Term-dependency of class c_4: $v_{c4} = \{(w_7, 4.444, 5.000), (w_6, 2.500, 2.500), (w_8, 1.406, 2.500)\}$
- Term-dependency of class c_5: $v_{c5} = \{(w_8, 4.444, 5.000), (w_5, 1.667, 2.500)\}$

6.1.3 Term-to-Term Relationship Mapping

Term-to-term relationship mapping is a further learning process that calculates the inter-relationships between every term in the term list of a class (the term-list of a class that has been created in the term-to-class relationship mapping process). In the term-to-class relationship mapping process, we find out the weighted relationship between a term and a class, but we do not know how those terms are related to each other inside the class. Therefore, the term-to-term relationship mapping further finds out and calculates this weighted relationship between those terms. We calculate and create a directed relation between two terms, as shown in Figure 6.7.

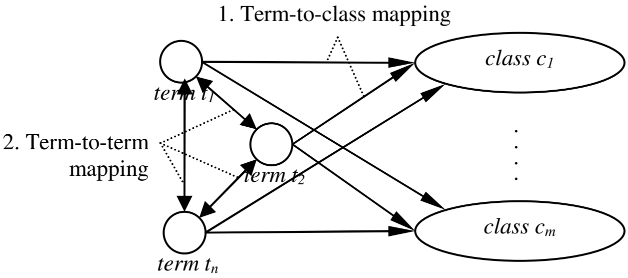


Fig. 6.7 Terms mapping to each other

To measure term-to-term relationship, we first select a certain number of terms in each class. In a real case, we determine a threshold *k* for the maximum number of highest ranked positive terms inside a term-dependency vector of each class to represent the term group of the corresponding class for calculation:

- *k*-number of ranked positive terms in each class: $V=\{v_1, v_2,\ldots,v_m\}$ *for each* $v_i = \{ (t_1, \chi^2_{t_1,c_j}, R_{t_1,c_j}), (t_2, \chi^2_{t_2,c_j}, R_{t_2,c_j}), \ldots, (t_k, \chi^2_{t_k,c_j}, R_{t_k,c_j}) \}$ *where* $R_{w_i,c_j} > 1$
- If the number of positive terms ($R_{t_i,c_j} > 1$) in a class is smaller than the threshold *k,* then we select all positive terms inside the class as the term group.

Example 6.6

Continued from Example 6.5, the selected term-group of each class, as represented in the following, for threshold $k = 4$

- Term group of class c_1 (4 terms selected): $v_{c1} = \{(t_2, 10.000, 3.333), (t_1, 0.476, 1.667), (t_3, 0.476, 1.667), (t_4, 0.476, 1.667)\}$
- Term group of class c_2 (2 terms selected): $v_{c2} = \{(t_3, 1.406, 2.500)\}$
- Term group of class c_3 (4 terms selected): $v_{c3} = \{(t_5, 3.750, 2.500), (t_6, 2.500, 2.000), (t_1, 1.406, 2.500), (t_4, 1.406, 2.500)\}$
- Term group of class c_4 (3 terms selected): $v_{c4} = \{(t_7, 4.444, 5.000), (t_6, 2.500, 2.000), (t_8, 1.406, 2.500)\}$
- Term group of class c_5 (2 terms selected): $v_{c5} = \{(t_8, 4.444, 5.000), (t_5, 1.667, 2.500)\}$

The relationship mapping process requires a document corpus (also a corpus of Chinese text documents) for learning purpose. In this term-to-term relationship mapping process, the document corpus is not required to be the same as the corpus that is used in the term-to-class relationship mapping process. Moreover, the document corpus is not required to be a classified corpus, because in this mapping process we are going to extract and find out the relationship between terms, so that the information of which class of a document refers to is unnecessary.

6.1.3.1 Term-to-Term Independency Measurement by χ^2

In the term-to-term relationship mapping process, we similarly apply the χ^2 statistical measurement of all the terms in the term-group v_i of each class c_i. The equation for χ^2 statistics is modified to measure the independency between two terms, instead of between a term and a class in the previous term-to-class mapping process. The co-occurrence frequencies between two terms - t_a and t_b are expressed in a modified two-way contingency table as shown in Table 6.18.

Table 6.18 2x2 term-to-term contingency table of term t_a and term t_b

	t_b	$\neg t_b$	Σ
t_a	O_{t_a,t_b}	$O_{t_a,\neg t_b}$	$O_{t_a,t_b} + O_{t_a,\neg t_b}$
$\neg t_a$	$O_{\neg t_a,t_b}$	$O_{\neg t_a,\neg t_b}$	$O_{\neg t_a,t_b} + O_{\neg t_a,\neg t_b}$
Σ	$O_{t_a,t_b} + O_{\neg t_a,t_b}$	$O_{t_a,\neg t_b} + O_{\neg t_a,\neg t_b}$	$O_{t_a,t_b} + O_{t_a,\neg t_b} + O_{\neg t_a,t_b} + O_{\neg t_a,\neg t_b} = N$

The term-to-term contingency table is comprised of the cells of observed frequency $O_{i,j}$ where $i \in \{t_a, \neg t_b\}$ and $j \in \{t_b, \neg t_b\}$. Thus, O_{t_a,t_b} is the observed frequency (number) of documents which contain term t_a as well as term t_b; $O_{t_a,\neg t_b}$ is the observed frequency of documents which does not contain term t_a and also does not contain term t_b; $O_{\neg t_a,t_b}$ is the observed frequency of documents which does not contain term t_a but contain the term t_b; and $O_{\neg t_a,\neg t_b}$ is the observed frequency of documents which does not contain both terms t_a and t_b.

The observed frequency is compared to the expected frequency $E_{i,j}$ where $i \in \{t_a, \neg t_b\}$ and $j \in \{t_b, \neg t_b\}$. $E_{i,j}$ is defined as

$$E_{i,j} = \frac{\sum_{a \in \{t_a, \neg t_b\}} O_{a,j} \sum_{b \in \{t_a, \neg t_b\}} O_{i,b}}{N}$$

The χ^2 statistical independency measurement for term t and class c introduced in Chapter 3.3.2 are now modified as follows, which measure the dependency between two terms t_a and t_b, instead of measuring between a term t and a class c.

$$\chi^2_{t_a,t_b} = \sum_{i \in \{t_a, \neg t_a\}} \sum_{j \in \{t_b, \neg t_b\}} \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}}$$

Note that by this term-to-term independency measurement, $\chi^2_{t_a,t_b} \neq \chi^2_{t_b,t_a}$, and $\chi^2_{t_a,t_b} = \chi^2_{t_a,t_i}$ *if* $a = b$. Normalization is also applied to every term-to-term independency value by the ratio of $\chi^2_{t_a,t_b}$ value and term-to-class independency value ($\chi^2_{t_a,c_i}$). The normalization of terms t_a and t_b is defined as:

$$n\chi^2_{t_a,t_b} = \frac{\chi^2_{t_a,t_b}}{\chi^2_{t_a,c_i}}$$

After the term-to-term relationship mapping and normalization process, we can obtain a $k \times k$ term-to-term dependency matrix containing the value of $n\chi^2_{t_a,t_b}$ and R_{t_a,t_b} as shown in Tables 6.19 and 6.20. The tables contain the term independency values representing the relationship of every term-to-term pair within a class.

Table 6.19 Term dependency $n\chi^2_{t_a,t_b}$ of class c_i

	t_1	t_2	…	t_{k-1}	t_k
t_1	*1*	$n\chi^2_{t_2,t_1}$	…	$n\chi^2_{t_{k-1},t_1}$	$n\chi^2_{t_k,t_1}$
t_2	$n\chi^2_{t_1,t_2}$	*1*	…	$n\chi^2_{t_{k-1},t_2}$	$n\chi^2_{t_k,t_2}$
…	…	…	…	…	…
t_{k-1}	$n\chi^2_{t_1,t_{k-1}}$	$n\chi^2_{t_2,t_{k-1}}$	…	*1*	$n\chi^2_{t_k,t_{k-1}}$
t_k	$n\chi^2_{t_1,t_k}$	$n\chi^2_{t_2,t_k}$	…	$n\chi^2_{t_{k-1},t_k}$	*1*

Table 6.20 Term dependency R_{w_a,w_b} of class c$_i$

	t_1	t_2	…	t_{k-1}	t_k
t_1	R_{t_1,t_1}	R_{t_2,t_1}	…	R_{t_{k-1},t_1}	R_{t_k,t_1}
t_2	R_{t_1,t_2}	R_{t_2,t_2}	…	R_{t_{k-1},t_2}	R_{t_k,t_2}
…	…	…	…	…	…
t_{k-1}	$R_{t_1,t_{k-1}}$	$R_{t_2,t_{k-1}}$	…	$R_{t_{k-1},t_{k-1}}$	$R_{t_k,t_{k-1}}$
t_k	R_{t_1,t_k}	R_{t_2,t_k}	…	R_{t_{k-1},t_k}	R_{t_k,t_k}

Example 6.7

The result of term-to-class relationship vectors of each class $C=\{c_1, c_2, c_3, c_4, c_5\}$ from Example 6.4 is shown as follows:

- $v_{c1} = \{(t_1, 0.476, 1.667), (t_2, 10, 3.333), (t_3, 0.476, 1.667), (t_4, 0.476, 1.667), (t_5, 0.079, 0.833), (t_6, 4.286, 0), (t_7, 0.476, 0), (t_8, 1.071, 0)\}$
- $v_{c2} = \{(t_1, 0.625, 0), (t_2, 1.071, 0), (t_3, 1.406, 2.500), (t_4, 0.625, 0), (t_5, 1.667, 0), (t_6, 0, 1.000), (t_7, 0.278, 0), (t_8, 0.625, 0)\}$
- $v_{c3} = \{(t_1, 1.406, 2.500), (t_2, 1.071, 0), (t_3, 0.625, 0), (t_4, 1.406, 2.500), (t_5, 3.750, 2.500), (t_6, 2.500, 2.000), (t_7, 0.278, 0), (t_8, 0.625, 0)\}$
- $v_{c4} = \{(t_1, 0.625, 0), (t_2, 1.071, 0), (t_3, 0.625, 0), (t_4, 0.625, 0), (t_5, 1.667, 0), (t_6, 2.500, 2.000), (t_7, 4.444, 5.000), (t_8, 1.406, 2.500)\}$
- $v_{c5} = \{(t_1, 0.278, 0), (t_2, 0.476, 0), (t_3, 0.278, 0), (t_4, 0.278, 0), (t_5, 1.667, 2.500), (t_6, 1.111, 0), (t_7, 0.123, 0), (t_8, 4.444, 5)\}$

The result of selected term-group for each class, by selecting top k ranked positive terms (*for* $k = 4$):

- $v_{c1} = \{(t_2, 10, 3.333), (t_1, 0.476, 1.667), (t_3, 0.476, 1.667), (t_4, 0.476, 1.667)\}$
- $v_{c2} = \{(t_3, 1.406, 2.500)\}$
- $v_{c3} = \{(t_5, 3.750, 2.500), (t_6, 2.500, 2.000), (t_1, 1.406, 2.500), (t_4, 1.406, 2.500)\}$
- $v_{c4} = \{(t_7, 4.444, 5.000), (t_6, 2.500, 2.000), (t_8, 1.406, 2.500)\}$
- $v_{c5} = \{(t_8, 4.444, 5), (t_5, 1.667, 2.500)\}$

The first step is to retrieve the term-to-class relationship vector and create a term-group containing at most four highest ranked terms for each class, as shown in Table 6.21.

Table 6.21 Term-groups created for classes c_1 to c_5

class	v_{c_i}
c_1	t_2, t_1, t_3, t_4
c_2	t_3
c_3	t_5, t_6, t_1, t_4
c_4	t_7, t_6, t_8
c_5	t_8, t_5

As stated in the process description, a document corpus is required for learning the term-to-term relationship mapping. The new document corpus can be different from that which has been used in the term-to-class relationship mapping, and the new document corpus needs not be a classified corpus (i.e. all unlabeled documents). The details of the document corpus used in this example are shown as follows (Table 6.22):

- Document corpus (number of documents = 10): $D_2 = \{d_{1,} d_2,\ldots, d_{10}\}$
- Unlabeled classes (not required in this process)
- Extracted candidate terms (distinct terms in all created term-group): $t_1,\ldots, t_8$

Table 6.22 Content of terms and document distribution of the document corpus D_2

Document	Term occurrence	Class
d_1	t_1, t_2	-
d_2	t_1, t_2	-
d_3	t_1, t_2, t_3, t_4, t_5	-
d_4	t_3, t_4, t_5	-
d_5	t_6, t_8	-
d_6	t_3, t_4, t_5, t_6	-
d_7	t_1, t_5, t_6	-
d_8	t_3, t_6	-
d_9	t_6, t_7, t_8	-
d_{10}	t_5, t_6, t_8	-

Every term-group of each class requires a separate term-to-term relationship mapping learning. This means that if there is n number of classes in the class vector C, there requires n separated learning processes for generating all term-to-term relationship mappings. For example, there are 5 classes (5 term-groups) in the Example 6.7 as shown in Table 6.21, so there requires 5 separated term-to-term independency measurements for each term-group.

In this learning process, every term in the term-group is first transformed as a "class-label" for processing the χ^2 based term-to-term independency measurement. Then each "class" is further mapped to a set of documents containing the term (the "class-label"), as shown in Figure 6.8.

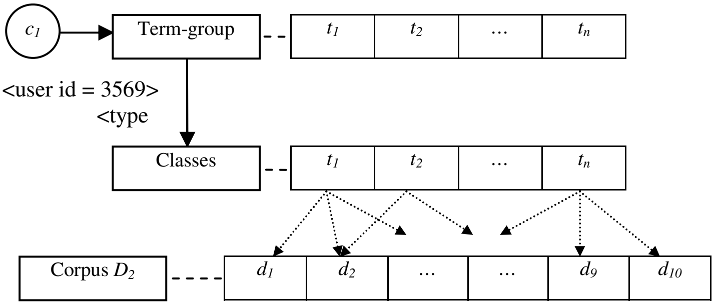


Fig. 6.8 Create document links to each term in the term-group

Steps of term-to-term relationship mapping process

```
STEP 1: Retrieve the term-to-class relationship vectors $V$
        For each $v$ in $V$
            Rank every term $t$ in v by $\chi^2_{t,c}$
            Select top $k$-number of w as a term-group
            $T = \{t_1, t_2,\ldots,t_k\}$
        Next
STEP 2: Prepare a unlabeled document corpus $D_2$
                        $D = \{d_1, d_2,\ldots\ldots\}$
STEP 3: Create new class vector $v$ with $k$-number of terms in each class
                        $V = \{v_{c1}, v_{c2},\ldots,v_{cm}\}$
STEP 4: Transform terms in term-group to class
                        $C = \{t, t,\ldots\ldots\}$
STEP 5: Retrieve and link documents from $D_2$ to each “class-label”
        For each $w$in $C$
            For each $d$ in $D_2$
                If $d$ contains $t$
                    Link $d$ to class $t$
                End If
            Next
        Next
STEP 6: Independency measurement for every term-pair
        For every term-to-class relationship vector $v$
            Create $C$ for $v$
            For each $t_a$ in $C$
                For each $t_b$ in $C$
                  calculate $\chi^2_{t_a,t_b} = \sum_{i\in\{t_a,\neg t_a\}} \sum_{j\in\{t_b,\neg t_b\}} \frac{(O_{i,j}-E_{i,j})^2}{E_{i,j}}$
                  calculate $R_{t_a,t_b} = \frac{O_{t_a,t_b}}{E_{t_a,t_b}}$
                  normalize $n\chi^2_{t_a,t_b} = \frac{\chi^2_{t_a,t_b}}{\chi^2_{t_a,c_i}}$
             Next
          Next
       Next
```

Fig. 6.9 Term-to-term relationship mapping steps

An example of the above step and its detailed independency measure is illustrated in the following (Example 6.8).

Example 6.8

The example illustrates the calculation of terms mapping in class c_1, the data is represented as follows (continued from Example 6.7):

- Document corpus (number of documents = 10): $D_2 = \{d_{1,} d_2,..., d_{10}\}$
- Term-to-class relationship vector of class c_1: $v_{c1} = \{(t_2, 10, 3.333), (t_1, 0.476, 1.667), (t_3, 0.476, 1.667), (t_4, 0.476, 1.667)\}$
- Transform term-group to class-label: $C_{c1} = \{t_2, t_1, t_3, t_4\}$
- Link up documents from D_2 to C_{c1} (result shown in Tables 6.23 and 6.24)

Table 6.23 Document link from D_2 to C_{c1}

Class	Document	Document count
t_2	d_1, d_2, d_3	3
t_1	d_1, d_2, d_3, d_7	4
t_3	d_3, d_4, d_6, d_8	4
t_4	d_3, d_4, d_6	3

Table 6.24 Term-to-term table (10 documents for the term-group of class c_1)

	t_2	t_1	t_3	t_4
t_2	d_1, d_2, d_3	d_1, d_2, d_3	d_3	d_3
t_1	d_1, d_2, d_3	d_1, d_2, d_3, d_7	d_3	d_3
t_3	d_3	d_3	d_3, d_4, d_6, d_8	d_3, d_4, d_6
w_4	d_3	d_3	d_3, d_4, d_6	d_3, d_4, d_6

All the following tables (Tables 6.25, 6.26, 6.27, and 6.28) are observed from Tables 6.23 and 6.24 and prepared for calculating the relationship of term t_1 to other terms (t_1, t_2, t_3, t_4). The term-to-term relationship mapping step and the calculation are shown as follows:

Calculating t_1 and t_1:

Table 6.25 2x2 term-to-term contingency table of term t_1 and term t_1

	t_1	$\neg t_1$	Σ
t_1	4	0	4
$\neg t_1$	0	3	3
Σ	4	3	7

The observed frequency: $O_{t_1,t_1} = 4$, $O_{t_1,\neg t_1} = 0$, $O_{\neg t_1,t_1} = 4$, $O_{\neg t_1,\neg t_1} = 3$. The expected frequency: $E_{t_1,t_1} = 2.291$, $E_{t_1,\neg t_1} = 1.71$, $E_{\neg t_1,t_1} = 1.71$, $E_{\neg t_1,\neg t_1} = 1.29$. The dependency value of t_1 and t_1: $\chi^2_{t_1,t_1} = 7$.

Calculating t_1 and t_2:

Table 6.26 2x2 term-to-term contingency table of term t_1 and term t_2

	t_2	$\neg\ t_2$	Σ
t_1	*3*	*1*	*4*
$\neg\ t_1$	*0*	*3*	*3*
3	*3*	*4*	*7*

The observed frequency: $O_{t_1,t_2} = 3$, $O_{t_1,\neg t_2} = 1$, $O_{\neg t_1,t_2} = 0$, $O_{\neg t_1,\neg t_2} = 3$. The expected frequency: $E_{t_1,t_2} = 1.71$, $E_{t_1,\neg t_2} = 2.29$, $E_{\neg t_1,t_2} = 1.29$, $E_{\neg t_1,\neg t_2} = 1.71$. The dependency value of t_1 and t_2: $\chi^2_{t_1,t_2} = 3.938$.

Calculating t_1 and t_3:

Table 6.27 2x2 term-to-term contingency table of term t_1 and term t_3

	t_3	$\neg\ t_3$	Σ
t_1	*1*	*3*	*4*
$\neg\ t_1$	*3*	*0*	*3*
Σ	*4*	*3*	*7*

The observed frequency: $O_{t_1,t_3} = 1$, $O_{t_1,\neg t_3} = 3$, $O_{\neg t_1,t_3} = 3$, $O_{\neg t_1,\neg t_3} = 0$. The expected frequency: $E_{t_1,t_3} = 2.29$, $E_{t_1,\neg t_3} = 1.71$, $E_{\neg t_1,t_3} = 1.71$, $E_{\neg t_1,\neg t_3} = 1.29$. The dependency value of t_1 and t_3: $\chi^2_{t_1,t_3} = 3.938$.

Calculating t_1 and t_4:

Table 6.28 2x2 term-to-term contingency table of term t_6 and term t_4

	t_4	$\neg\ t_4$	Σ
t_1	*1*	*3*	*4*
$\neg\ t_1$	*2*	*1*	*3*
Σ	*3*	*4*	*7*

The observed frequency: $O_{t_1,t_4}=1$, $O_{t_1,\neg t_4}=3$, $O_{\neg t_1,t_4}=2$, $O_{\neg t,\neg t_4}=1$. The expected frequency: $E_{t_1,t_4}=1.71$, $E_{t_1,\neg t_4}=2.29$, $E_{\neg t_1,t_4}=1.29$, $E_{\neg t_1,\neg t_4}=1.71$. The dependency value of t_1 and t_4: $\chi^2_{t_1,t_4}=1.215$.

The dependency values of t_1-t_1, t_1-t_2, t_1-t_3, and t_1-t_4 are thus calculated and shown in the first row in Table 3.30. To complete all dependency values between all terms in class c_1 including (second row): t_2-t_1, t_2-t_2, t_2-t_3, t_2-t_4, (third row): t_3-t_1, t_3-t_2, t_3-t_3, t_3-t_4, (forth row): t_4-t_1, t_4-t_2, t_4-t_3, t_4-t_4, there requires three more calculation steps similarly to that had been shown in the above example. All dependency values are thus calculated and shown in Table 6.29. The final result of term-to-term relationship mapping and its dependency values are shown in Tables 6.30 and 6.31.

Table 6.29 χ^2 term-to-term mapping of 4 terms in class c_1

	t_1	t_2	t_3	t_4
t_1	*7.000*	*3.938*	*3.938*	*1.215*
t_2	*4.800*	*8.000*	*0.533*	*0.036*
t_3	*3.938*	*1.125*	*7.000*	*3.928*
t_4	*0.533*	*0.036*	*4.800*	*8.000*

Table 6.30 Final Result - terms dependency $n\chi^2_{t_a,t_b}$ of class c_1

$n\chi^2_{t_a,t_b}$	t_1	t_2	t_3	t_4
t_1	*1*	*0.5626*	*0.5626*	*0.1736*
t_2	*0.6000*	*1*	*0.0666*	*0.0045*
t_3	*0.5626*	*0.1607*	*1*	*0.5611*
t_4	*0.0666*	*0.0045*	*0.6000*	*1*

Table 6.31 Final Result - terms dependency R_{t_a,t_b} of class c_1

R_{t_a,t_b}	t_1	t_2	t_3	t_4
t_1	*1.077*	*1.556*	*0.389*	*0.519*
t_2	*1.312*	*1.273*	*0.438*	*1.556*
t_3	*0.398*	*0.549*	*1.077*	*1.556*
t_4	*0.438*	*0.583*	*1.312*	*1.273*

The tabular representation of the term dependency can be converted into a directed Ontology Graph: OG = (V, E) where V is the set of vertices of terms, $V = \{t_1, t_2,..., t_{k-1}, t_k\}$, and A is the set of directed and weighted edge: $E = \{(t_1, t_1, R_{t_1,t_1}),(t_1, t_2, R_{t_1,t_2}),..., (t_k, t_{k-1}, R_{t_k,t_{k-1}}),(t_k, t_k, R_{t_k,t_k})\}$ where $R_{t_a,t_b} > 1$. In the example 3.8, for $k = 4$, the visualized Ontology Graph is created as shown in Figure 6.10.

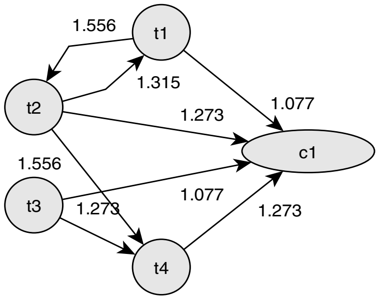


Fig. 6.10 Ontology Graph created for 4 terms in class c_1 *(k=4)*

The vertices in the graph are the top k terms in the class, and the edges in the graph are the directed and weighted link between two terms if their dependency relation is positive ($R_{t_a,t_b} > 1$). If the dependency relation of two concepts is negative ($R_{t_a,t_b} < 1$), the link is not created in the graph.

6.1.4 Concept Clustering

The concept clustering is the process of grouping semantically similar concepts into a tight semantic group. The directed interdependency graph created in the previous step is the base input for the concept clustering process. The idea is to group concepts with high weighted relations into a sub graph while separating out other concepts to create a new sub graph of low weighted relations. Clusters are automatically created without explicitly defining the number of clusters needs to be created. The highest weighted edge e_x with two vertices t_a and t_b is first grouped together to form an initial cluster. We then select the next highest weighted edge e_y with the next two vertices t_c and t_d. If the next selected vertices are linked by any vertices from the existing cluster, the vertices are put into that cluster. Otherwise a new cluster is formed with the inclusion of the selected vertices. The algorithm and clustering steps are shown in Figure 6.11. The result is an Ontology Graph containing several concept clusters, as shown in Figure 6.12.

Steps of term-to-concept clustering process

```
For every Ontology Graph $OG$
    Select the highest weighted edge $e_x = \{(t_a, t_b, R_{t_a,t_b})$ in vector $E$
    Create the first concept cluster containing $t_a$ and $t_b$
    For every edge $e$ in the edge vector $E$
        Select the next highest weighted edge $e_x = \{(t_c, t_d, R_{t_c,t_d})$
        If $t_c$ or $t_d$ appears in the existing cluster
            Put both $t_c$ and $t_d$ into that existing cluster
        Otherwise
            Create a new cluster containing $t_c$ or $t_d$ which does not appear
            in existing clusters
        End If
    Next $e$
Next $OG$
```

Fig. 6.11 Terms-to-concept clustering steps

The concept clustering process creates the second taxonomical relationship. The first taxonomical concept relationship is created in term-to-class relationship mapping, where all the terms in a single class are now further clustered and create a second layer of hierarchy. So that every concept cluster creates relationships to their related class as a parent, and then it creates relationships to all their contained terms as children. Finally the process creates a three-level taxonomical relationship in the Ontology Graph for the Example 6.8 (Figure 6.12).

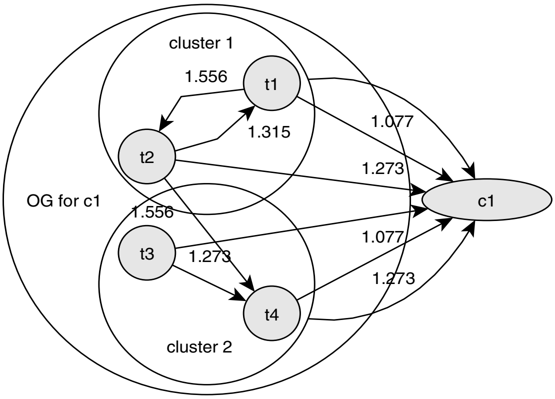


Fig. 6.12 Final Ontology Graph created for class c_1

Experiment 6.1

This experiment considers all the ontology learning process as described in this chapter. The experiment is focused on learning Ontology Graph in Chinese text, and it gives the experimental results in each step to show the effectiveness of the whole ontology learning process.

Experimental data setup (prepare the document corpus)

Details of the learning Chinese text document corpus D_1

- Document corpus (number of documents = 2814): $D_1 = \{d_1, d_2, \ldots, d_{2814}\}$
- Average number of characters in each document: *965* (Chinese character)
- Labeled classes (number of classes = 10): $C = \{c_1, c_2, \ldots, c_{10}\}$

Table 6.32 Class label (Chinese & English) in document corpus D_1

Class	Class Label	(English)
c_1	文藝	Arts and Entertainments
c_2	政治	Politics
c_3	交通	Traffic
c_4	教育	Education
c_5	環境	Environment
c_6	經濟	Economics
c_7	軍事	Military
c_8	醫療	Health and Medical
c_9	電腦	Computer and Information Technology
c_{10}	體育	Sports

The ten topic classes are the class-label used in term-to-class relationship learning process. The document distribution in the ten classes is shown in Table 4.2:

Table 6.33 Document distribution of the ten classes (D_1)

Class	Document count
文藝	248
政治	505
交通	214
教育	220
環境	201
經濟	325
軍事	249
醫療	204
電腦	198
體育	450
Total	2814

The documents of the corpus in every class are further divided into 70% for the learning set (D_1*-Learn*), and 30% for the testing test (D_1*-Test*), as shown in Table 4.3. We use only the 70% classified documents (1972 documents) for the process of term extraction and term-to-class mapping.

Table 6.34 Document distribution for learning and testing

Class	D_1*-Learn* (70%)	D_1*-Test* (30%)
文藝	174	74
政治	354	151
交通	150	64
教育	154	66
環境	141	60
經濟	228	97
軍事	174	75
醫療	143	61
電腦	139	59
體育	315	135
Total	1972 (70% of 2814)	842 (30% of 2814)

There is another Chinese document set from an unclassified corpus (D_2), as shown in Table 4.4. It is used for the process of term-to-term mapping and concept clustering. The corpus D_2 contains a relatively large amount of documents (57218 documents), which is collected from a Chinese News web site (人民網 / *www.people.com.cn*), with an average of 2349 Chinese characters in each news document.

Table 6.35 Documents distribution of corpus D_2

(Unclassified)	Document count
人民網 News	57218
Total	57218

Term extraction for the Ontology Graph learning process

Data of the word extraction process:

- Learning document corpus (number of documents = 1972) D_1*-Learn* = $\{d_1, d_2,\ldots,d_{1972}\}$
- Labeled classes (number of classes = 10): $C = \{c_1, c_2,\ldots,c_{10}\}$ (refer to Table 6.33)
- Extracted candidate terms (number of terms = 35840): $T = \{t_1, t_2,\ldots, t_{35840}\}$

Table 6.36 Statistics of term extraction results

Statistic	Count
Number of documents	1972
Number of classes	10
Minimum document size in class	139
Maximum document size in class	354
Number of unique term extracted	35840

Statistic of Term Extracted in the Ontology Graph learning process

The candidate term list extracted from the previous step is then processed with term-to-class relationship mapping. The dependency value of every term-to-class is measured by χ^2 and either a positive or a negative dependency is measured by R. The results of the number of positive and negative terms in the ten classes are shown in Table 6.37.

Table 6.37 Results of term-to-class relationship mapping

Class	Number of positive terms	Number of negative terms
文藝	867	29281
政治	966	37481
交通	769	34691
教育	904	30604
環境	788	32823
經濟	664	35680
軍事	727	33439
醫療	862	35527
電腦	774	30671
體育	956	37061

The corresponding ratio between positive and negative terms is also shown in Table 6.38. This result shows that the term-to-class relationship mapping successfully selects the top 3% of relevant (positive) terms in each class while it is able to filter out 97% irrelevant (negative) terms in each class.

The distribution of the selected relevant terms to the four grammatical categories, POS (nouns, verbs, adjectives, and adverbs) is shown in Figure 6.1.3. The figure shows that the highest number of terms selected are noun (51%), followed by verb (28%), adjective (14%), and adverb (3%). The result shows that noun is the most relevant term to build ontology concepts, because noun terms are mostly dependent on a class. Adverb and others (conjunction, preposition, number, etc.) are therefore necglectable because they have less dependency on classes.

Table 6.38 Ratio of positive to negative terms in each class

Class	Ratio of Positive: negative terms
文藝	0.0288 : 0.9712
政治	0.0320 : 0.9749
交通	0.0255 : 0.9783
教育	0.0300 : 0.9713
環境	0.0261 : 0.9766
經濟	0.0220 : 0.9817
軍事	0.0241 : 0.9787
醫療	0.0286 : 0.9763
電腦	0.0257 : 0.9754
體育	0.0317 : 0.9749
Average	0.0275 : 0.9725

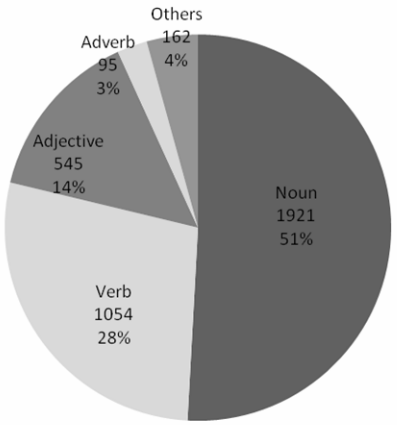


Fig. 6.13 The distribution of terms in their POS

6.1.5 Sample Result of Domain Ontology Graph Generation (10 Domains)

Figures 6.14 to 6.23 visualize the generated Domain Ontology Graphs (*DOG*) of the 10 classes (domains). The learning process selects 30 highest ranked positive terms in each class (*k=30)* to generates the corresponding Ontology Graph. The figures only visualize the terms and their relationships. The detailed results of term-to-class relationship mapping (Tables A.1 – A.10) and term-to-term relationship mapping (Tables A.11 – A.20) of those 30 terms (the corresponding list of English translation is provided in the Appendix) in each class are shown in the Appendix.

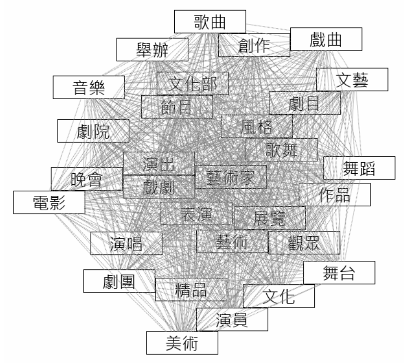


Fig. 6.14 *DOG* (文藝 Arts and Entertainments)

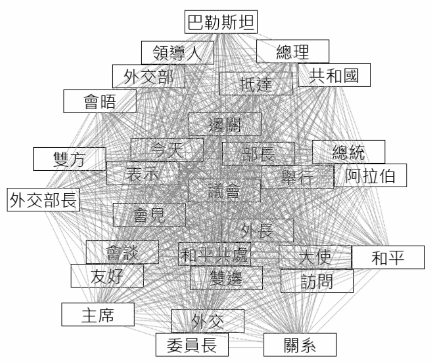


Fig. 6.15 *DOG* (政治 Politics)

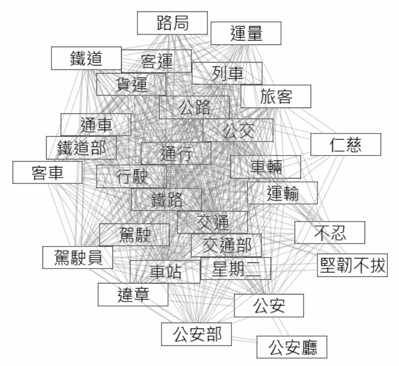


Fig. 6.16 *DOG* (交通 Traffic)

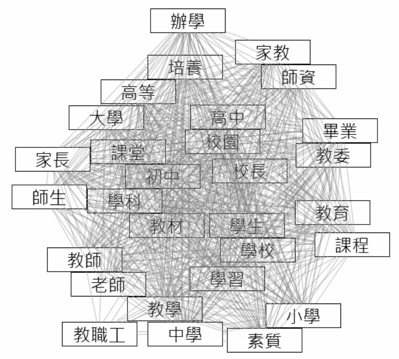


Fig. 6.17 *DOG* (教育 Education)

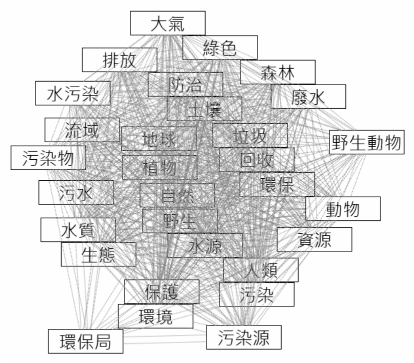


Fig. 6.18 *DOG* (環境 Environment)

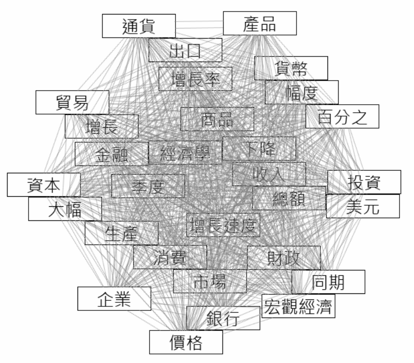


Fig. 6.19 *DOG* (經濟 Economics)

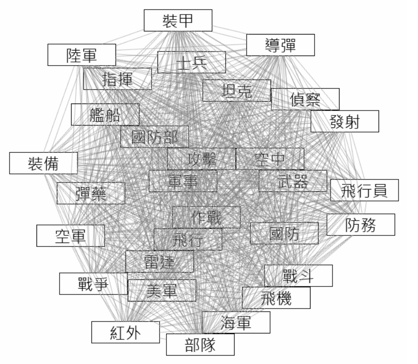


Fig. 6.20 *DOG* (軍事 Military)

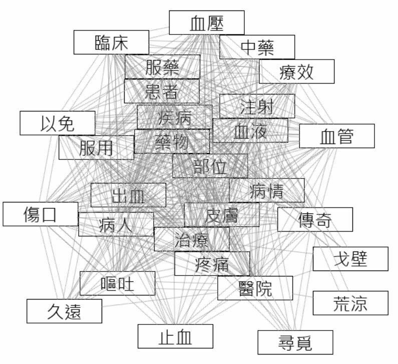


Fig. 6.21 *DOG* (醫療 Health and Medical)

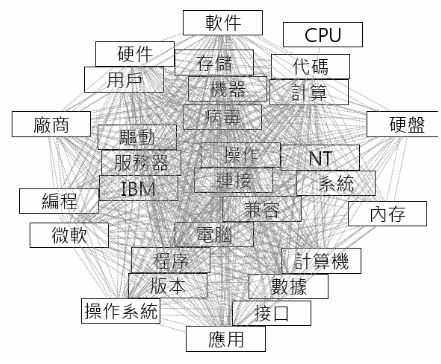


Fig. 6.22 *DOG* (電腦 Computer and Information Technology)

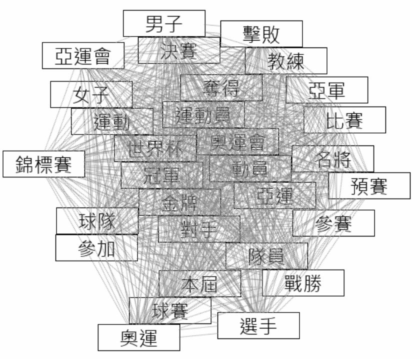


Fig. 6.23 *DOG* (體育 Sports)

Chapter 7
Ontology Graph Generation Process

Abstract. In this chapter, we define an ontology generation method that transforms the ontology learning outcome to the Ontology Graph format for machine processing and also can be visualized for human validation. We first formalize the Ontology Graph structure and define the generation methods. After that an experiment of automatic generation of Domain Ontology Graph with the visualized results is presented.

7.1 Ontology in Information Systems

The KnowledgeSeeker provides an ontology modeling framework for intelligent information system based on Chinese text. A typical information system on text such as content management system, web news portal contains a large amount of text documents. These information systems can be described by three different forms based on their degree of structure on managing the text data (Rifaieh and Benharket 2006):

- Highly Informal – the text data are stored loosely in natural language as its original without any pre-processing and analyzing on the text.
- Semi Informal – the text data are processed and expressed in a more structural way such as term index and taxonomy, which have been used in many traditional IR systems.
- Semi Formal – the text data are processed and expressed formally in a structured format such as XML, XML Schema, WSDL, and Topic Map. It enhances data integrity and information sharing.
- Highly Formal – the text data are processed and expressed in logic-based languages such as FOL, RDF, and OWL. The data is also enhanced with a computable knowledge such as Ontology for intelligent and semantic processing.

A formal text based information system highly relies on Ontology, and there are three different types of Ontology defined according to its level of abstraction (Figure 7.1).

- Upper Ontology – also called top-level ontology that is universal, generic enough to model common sense knowledge. It is generic and domain independent.
- Domain Ontology – ontology created for a specific domain or particular area of interest such as science domain or entertainment domain. This ontology can be extended from upper ontology.

E.H.Y. Lim, J.N.K. Liu, and R.S.T. Lee: Knowledge Seeker, ISRL 8, pp. 99–119.
springerlink.com

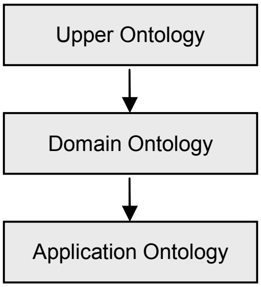


Fig 7.1 Three types of Ontology according to their level of abstraction

- Application Ontology – ontology created for used in specific application such as news service and intelligent agent application.

7.2 Ontology Graph Generation Process in KnowledgeSeeker

Ontology Graph is the ontology modeling format in KnowledgeSeeker system, and it is the knowledge representation used for intelligent information application development. As described in Chapters 3.2 and 3.3, the Ontology Graph is able to model concepts that are based on Chinese terms and their interdependency relationship, through the automatic ontology learning process. The Ontology Graph is created as a graphical structure with vertices and edges between them. In Chapter 3.3, we introduced the method of learning Ontology Graph for a class (a domain), so that we can define that Ontology Graph as a Domain Ontology Graph (*DOG*), which is used as the middle layer (domain ontology) between upper ontology and application ontology as shown in Figure 7.2.

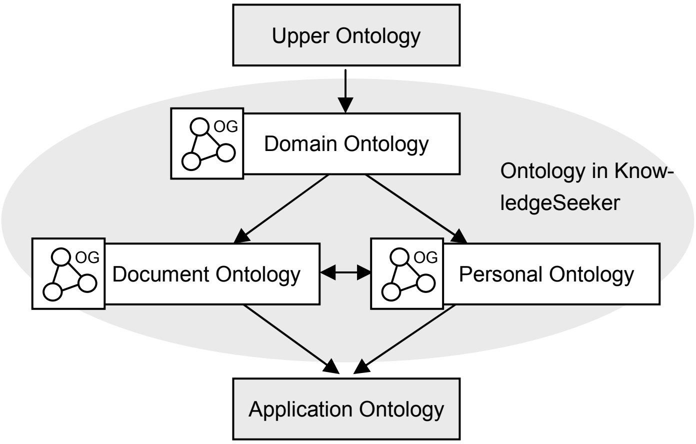


Fig 7.2 Types of ontology in different levels

In addition to the three layers of ontology – upper ontology, domain ontology and application ontology, KnowledgeSeeker defines two additional types of ontologies, they are document ontology and personal ontology. These two types of ontology are created based on the domain ontology, and it serves as a mediator in between the domain ontology and application ontology (Figure 7.2). The upper ontology and the application ontology are usually defined in existing ontology modeling languages, while the other three core ontologies in KnowledgeSeeker, including the domain ontology, document ontology and personal ontology are created in the form of Ontology Graph.

7.2.1 Definition of Ontology Graph Structure

In KnowledgeSeeker, we define Ontology Graph (*OG*) to model a set of concepts. Concepts are created by set of terms and relations between them. The relations of terms are enhanced by weights, which are generated automatically by the statistical learning method as presented in Chapter 6. In the following, we formalize the definition of *OG*:

Definition of OG

The Ontology Graph (*OG*) in KnowledgeSeeker system is defined as:

$$OG_d = <T, F, H, R, C, A>$$

- d defines the domain of the Ontology Graph is associated with
- T is a set of terms t_i of OG_d
- F is a set of word functions of terms $t_i \in T$
- H is a set of taxonomy relationships of T
- R is a set of relations between t_i and t_j, where t_i , $t_j \in T$
- C is a set of clusters of $t_i,...,t_n$, where $t_1,...,t_n \in T$
- A is a set of axioms that characterize each relation of R

Definition of Terms in OG

The term (*T*) in *OG* is symbol in the form of lexical words. The term itself does not define any concepts or semantic meanings, unless relations are assigned to it. In the natural language of Chinese, meaningful terms for human understanding are normally formed by 2-4 Chinese characters. A term t_i is assigned to the domain d with a weight $w_{t,d}$ in the Ontology Graph OG_d as the initial relation, refers to how much the term t_i is related to the domain d:

$$T_d(t_i, w_j) \text{ where } t_i \in T$$

Definition of Word function of terms in OG

Word functions are assigned to terms in *OG* to differentiate different kinds and nature about the terms. Word function *F* in *OG* is defined as:

$$F = (T, P, M_F)$$

- T is the set of terms t_i in OG_d
- P is a set of types of word function

- M_F is a set of mapping between a term t_i and word function p_i, where M_F is a mapping functions defined as:

$$M_F\ (t_i,\ p_i) \text{ where } t_i \in T,\ p_i \in P$$

The basic word functions include the following:

$$P \in \{N, CN, PN, ADJ, REF\_ADJ, EMO\_ADJ, VERB, REF\_VERB, EMO\_VERB\}$$

- Noun (*N*) – includes common noun and proper noun:
  - Common Noun (*CN*) - A term that refers and describes a person, place, thing, state, etc.
  - Proper Noun (*PN*) – A term that names people, places, and things.
- Adjective (*ADJ*) – includes referential adjectives and emotive adjectives
  - Referential adjectives (*REF_ADJ*) – Expressive terms that describe and modify the meaning of a noun.
  - Emotive adjectives (*EMO_ADJ*) – Cognitive terms that describe and modify the meaning of a noun.
- Verb (*VERB*) – includes referential verbs and emotive verbs
  - Referential verbs (*REF_VERB*) – Objective terms that describe an action or a state
  - Emotive verbs (*EMO_VERB*) – Subjective terms that describe an action of state

Definition of Hierarchy in OG

The hierarchy in *OG* is a special type of relationship that describes the taxonomy between two terms. It is defined that a term t_i semantically contains t_j if t_i is a super-ordinates of t_j, namely $t_i \supseteq t_j$. The hierarchy consists of one-to-many relationship structure (a super-ordinate relates to many sub-ordinates and one sub-ordinate relates to only one super-ordinate). The hierarchy *H* in *OG* is defined as:

$$H = (S,\ Rel_H)$$

- *S* is a sub-set of terms *T* of OG_d that representing the super-ordinate and sub-ordinate terms
- Rel_H is a set of directed and weighted hierarchical relations between a super-ordinate term t_i and a sub-ordinate term t_j with a weight value w. Rel_H is a ranking function which associates the terms:

$$Rel_H\ (t_i,\ t_j,\ w_{t_i,t_j}\ )$$

Definition of Relations in OG

The relation *R* in *OG* is any semantic relationship between two terms t_i and t_j, namely $t_i \times t_j$,. The relation *R* in *OG* is defined as:

$$R = (T,\ Rel_S)$$

- T is the set of terms t_i of OG_d
- Rel_S is a set of directed and weighted semantic relations between two terms t_i and t_j with a weight value w. Rel_S is a ranking function which associates with the terms:

$$Rel_S\,(t_i,\ t_j,\ w_{t_i,t_j}\ )$$

Definition of Cluster in OG

The cluster C in OG separates all terms t_i into several clusters. A cluster is formed by a group of terms that are semantically similar to each other, and it expresses a generalized concept as a group rather than an explicit term. The cluster C in OG is defined as:

$$C = (L,\ S,\ M_C)$$

- L is a set of labels representing the cluster
- S is a sub-set of terms T of OG_d
- M_C is a set of mappings between a label l_i and a term t_i, where MC is a mapping functions defined as:

$$M_C\ (l_i,\ t_i) \text{ where } l_i \in L,\ t_i, \in S$$

7.2.2 Domain Ontology Graph Generation Process

The Domain Ontology Graph (*DOG*) is created from a large classified Chinese corpus in the ontology learning process. The generation is a semi-automatic process. The main flow of the automatic process had been discussed in Chapter 6 and it is summarized in Figure 7.3.

As shown in Figure 7.3, the manual processes include defining the initial term list (can be obtained from existing dictionary), defining and mapping the types of word

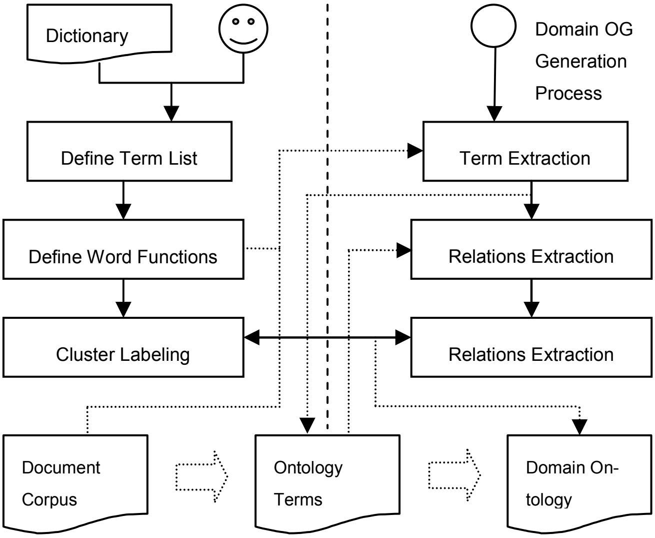


Fig. 7.3 Domain Ontology Graph generation process

function (also may be obtained from that dictionary), and labeling the concept clusters. The automatic processes include the domain terms extraction, terms relationship extraction (taxonomical and semantic relationship), and concept cluster extraction.

Example 7.1

Table 7.1 shows a term list of a sample *DOG* of the domain “*entertainment” (娛樂)* and Figure 7.4 visualizes the sample *DOG* as a directed graph. This example only shows a sample data and the data is not learnt by the automatic learning process.

The OG_d contains those definitions as in the following:

- *d = “娛樂”*
- *T = {(娛樂, 1), (音樂, 0.9), (電影, 0.9), (跳舞, 0.9), (流行舞, 0.9), (流行, 0.8), (爵士舞, 0.8), (爵士樂, 0.8), (經典, 0.7), (唱歌, 0.7), (戲院, 0.6), (卡通, 0.6), (動畫, 0.6), (記錄片, 0.5), (導演, 0.5), (演員, 0.5), (演出0.5), (歷史, 0.1)}*
- *F* = {(娛樂, CN), (音樂, CN), (電影, CN), (跳舞, CN), (流行舞, CN), (流行, CN), (爵士舞, CN), (爵士樂, CN), (經典, CN), (唱歌, REF_VERB), (戲院, CN), (卡通, CN), (動畫, CN), (記錄片, CN), (導演, CN), (演員, CN), (演出, REF_VERB), (歷史, d = “娛樂”
- T = {(娛樂, 1), (音樂, 0.9), (電影, 0.9), (跳舞, 0.9), (流行舞, 0.9), (流行, 0.8), (爵士舞, 0.8), (爵士樂, 0.8), (經典, 0.7), (唱歌, 0.7), (戲院, 0.6), (卡通*, 0.6), (動畫, 0.6), (記錄片, 0.5), (導演, 0.5), (演員, 0.5), (演出0.5), (歷史, 0.1)}*

Table 7.1 Term list of a sample *DOG* of domain “*entertainment*”

Term t_i	Term t_i (in English)	Word Function	Weight to Domain
娛樂	Entertainment	CN	1.0
音樂	Music	CN	0.9
電影	Movie	CN	0.9
跳舞	Dance	CN	0.9
流行舞	Pop Dance	CN	0.9
流行	Pop	CN	0.8
爵士舞	Jazz Dance	CN	0.8
爵士樂	Jazz Music	CN	0.8
經典	Classical	CN	0.7
唱歌	Sing	REF_VERB	0.7
戲院	Cinema	CN	0.6
卡通	Cartoon	CN	0.6
動畫	Anime	CN	0.6
記錄片	Documentary Film	CN	0.5
導演	Director	CN	0.5
演員	Actor	CN	0.5
演出	Perform	REF_VERB	0.5
歷史	History	CN	0.1

- *F = {(娛樂, CN), (音樂, CN), (電影, CN), (跳舞, CN), (流行舞, CN), (流行, CN), (爵士舞, CN), (爵士樂, CN), (經典, CN), (唱歌, REF_VERB), (戲院, CN), (卡通, CN), (動畫, CN), (記錄片, CN), (導演, CN), (演員, CN), (演出, REF_VERB), (歷史, CN)}*
- *H = {(娛樂, 音樂, 0.5), (娛樂, 電影, 0.5), (娛樂, 跳舞, 0.2), (跳舞, 流行舞, 0.6), (跳舞, 爵士舞, 0.6), (音樂, 流行, 0.5), (音樂, 爵士樂, 0.7), (音樂, 經典, 0.4), (電影, 卡通, 0.5), (電影, 動畫, 0.6), (電影, 記錄片, 0.3)}*
- *R = { (唱歌, 音樂, 0.8), (爵士舞, 爵士樂, 0.2), (流行舞, 流行, 0.6), (電影, 戲院, 0.8), (導演, 電影, 0.9), (演員, 電影, 0.9), (導演, 演出, 0.3), (演員, 演出, 0.9), (記錄片, 歷史, 0.7)}*
- C = { (音樂, 音樂), (電影, 電影), (跳舞, 跳舞), (跳舞, 流行舞), (音樂, 流行), (跳舞, 爵士舞), (音樂, 爵士樂), (音樂, 經典), (音樂, 唱歌), (電影, 戲院), (電影, 卡通), (電影, 動畫), (電影, 記錄片), (電影, 導演), (電影, 演員), (電影, 演出)}

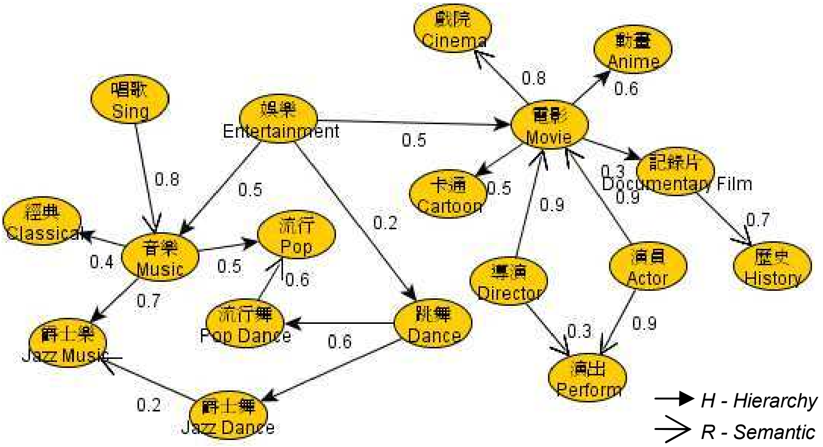


Fig. 7.4 A graphical representation of the *DOG* of "*entertainment*"

7.2.3 Document Ontology Graph Generation

A document ontology graph is extracted and generated from a document which is written in natural language, to express the ontological knowledge about the document. As natural language text contains unstructured knowledge which is only understood by human and it is hard to be processed by computer, a document ontology graph serves as a structured knowledge format to express the knowledge and meaning about a text in a computer processable format. This extraction and generation is called the *Text-to-OG* process.

Text-to-OG Process

The document ontology graph is used to convert a document of text to a graphical format. There are six steps to transform a text to *OG* as shown in Figure 7.5:

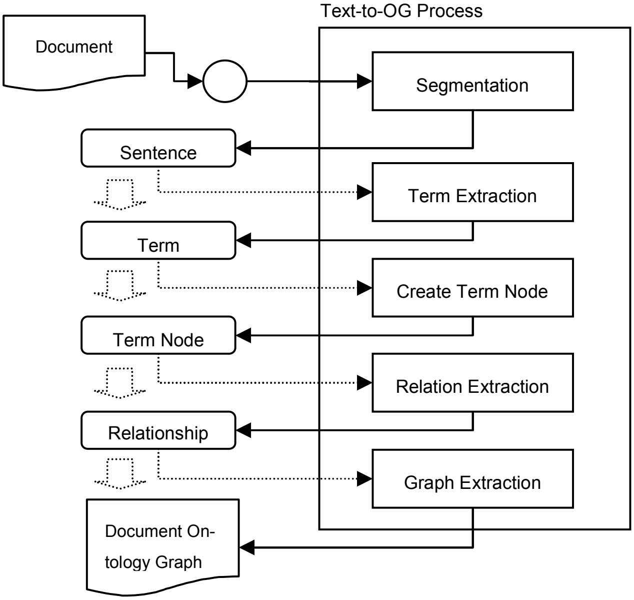


Fig. 7.5 Document Ontology Graph generation process

Components in Text-to-OG Process

1. Text – the document itself is written plainly by natural language, without any meta-data, markup, annotation, etc.
2. Sentence – the sentence is separated to express a concept normally.
3. Term – the term is segmented in a single sentence, consisting of multiple characters and being meaningful lexicons.
4. Term Node – the term node is the basic node expressed as a word in the Interdependency Graph.
5. Relationship – more than one term node related to each other and creating a relationship between nodes.
6. Document Ontology Graph – combining the extracted term nodes and their relationships from a document to have the Document Ontology Graph finally generated.

Text-to-OG Process Description

1. Divide text into sentences – the text is first divided into sentences for segmentation.
2. Segmentation of sentence into terms – the sentence is processed by segmentation algorithm such as the maximal matching algorithm to extract terms or word phrases from the sentence.
3. Create term nodes – term nodes are created for every meaningful word and word phrase, for the purpose of creating the interdependency graph.
4. Link all term nodes with relations – all the created terms nodes are linked with directed and weighted edges, to model the relationship between the terms.
5. Create relationship – the terms nodes with relations are created and extracted to relationship format.

6. Create Document Ontology Graph – the overall data is converted into *OG* definition for formalized Ontology Graph representation.

7.3 Automatic Generation of Domain Ontology Graph

The Domain Ontology Graph (*DOG*) generation module of KnowledgeSeeker is an automatic process which relies on labeled document corpus learning (refer to the ontology learning module discussed in Chapter 6). In this process, two additional threshold values are defined for generating a *DOG*. These two thresholds are used to control the size of the *DOG* to be generated. A larger size of *DOG* contains more terms and term-relations while a smaller size of *DOG* contains less of those. The first threshold value θ_u is set for the maximum number of terms which is selected in a class for the calculation of term dependency. The second threshold value θ_v is set for the minimum dependency values *(R)* in which the terms association is generated in the *DOG*. In summary, the thresholds are:

- θ_u – The maximum number of terms (nodes) in the Domain Ontology Graph
- θ_v – The minimum dependency value (edges) in the Domain Ontology Graph

The generation steps are shown in Figure 7.6:

Steps of Domain Ontology Graph (*DOG*) generation process

```
Obtain the term-list $T$ containing $\theta_u$ -number of  terms from ontology learning result
    $T = \{t_1, t_2,..., t_k\}$ where $k = \theta_u$
For every term $t$ in $T$
    Generates a Node $n_i$ in the DOG:
        Assign the node label by the term name
        Assign the node weight by the word-to-class dependency values ( $\chi^2_{w,c}$ )
    End for Node $n_i$
Next term
Obtain the term dependency values of the term-list $T$ from ontology learning result
For every term-term ($t_a – t_b$) dependency mapping
    If the dependency value $R$ is greater than or equal to $\theta_v$
        Generates an Edge $e_i$ between in the DOG:
            Associate with the two ends of the edge to the nodes of $t_a$ and $t_b$
            Set the edge weight to the word-to-word dependency value $\chi^2_{t_a,t_b}$
        End for Edge $e_i$
    End If
Next mapping
Remove all unlinked nodes in the DOG
```

Fig. 7.6 The steps of automatic domain ontology graph generation

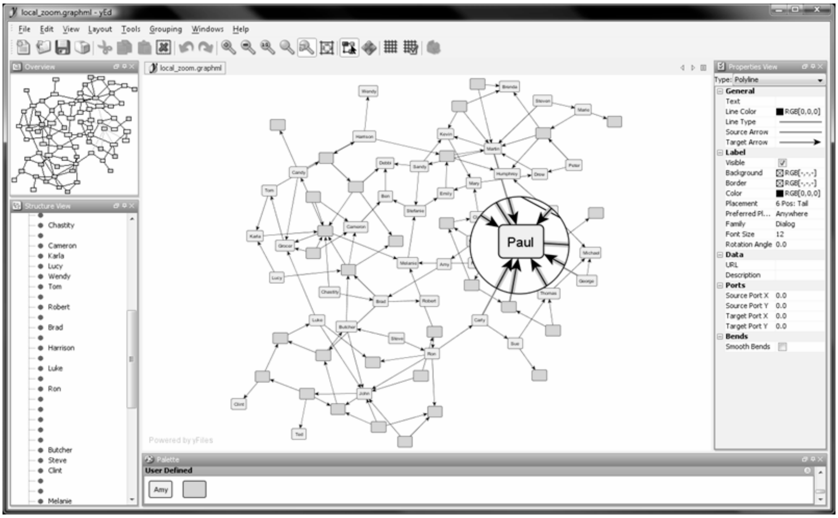
7.3.1 Experimental Setup

The objective of this Domain Ontology Graph (*DOG*) generation experiment is to observe the generated result through visualizing it in a graphical format. We select different values of the thresholds (θ_u and θ_v) as the variant parameters, where, θ_u ranges from 10 to 120, and θ_v ranges from 0 to 200. Every combination of the two parameters setting will generate one *DOG*. The threshold values used in the experiment are:

- θ_u *– 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120*
- θ_v *– 0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130, 140, 150, 160, 170, 180, 190, 200*

We reuse the experimental data setup and results from the ontology learning experiments described in Chapter 6 (Experiment 6.1). The data consists of a document corpus (D_1) with 2814 documents and 10 labeled classes (Table 6.32 – 6.33). The ontology learning result of the class "文藝 (*Arts and Entertainments)*" is used here to generate the *DOG* automatically.

The *DOG* generation program is implemented by Java and each *DOG* result is generated and written in a GraphML document, an XML-based file format for graphs (GraphML 2007, yFile XML Extension 2009). The GraphML document is further visualized by the software yED (Figure 7.7), a graph editor that is able to visualize a graph and to apply automatic layouts of the graph (yED 2009). An example of a generated GraphML document of an ontology graph is shown in Figure 7.8.



Type: Polyline

General	
Text	
Line Color	■ RGB[0,0,0]
Line Type	
Source Arrow	
Target Arrow	
Label	
Visible	☑
Background	☒ RGB[-,-,-]
Border	☒ RGB[-,-,-]
Color	■ RGB[0,0,0]
Placement	6 Pos: Tail
Preferred Pl...	Anywhere
Family	Dialog
Font Size	12
Rotation Angle	0.0
Data	
URL	
Description	
Ports	
Source Port X	0.0
Source Port Y	0.0
Target Port X	0.0
Target Port Y	0.0
Bends	
Smooth Bends	□

Fig. 7.7 The interface of yED Graph Editor (yED 2009)

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<graphml xmlns="http://graphml.graphdrawing.org/xmlns" xmlns:xsi=
http://www.w3.org/2001/XMLSchema-instance"
xmlns:y="http://www.yworks.com/xml/graphml"
xsi:schemaLocation="http://graphml.graphdrawing.org/xmlns
http://www.yworks.com/xml/schema/graphml/1.1/ygraphml.xsd">
  <key for="graphml" id="d0" yfiles.type="resources"/>
  <key attr.name="url" attr.type="string" for="node" id="d1"/>
  <key attr.name="description" attr.type="string" for="node" id="d2"/>
  <key for="node" id="d3" yfiles.type="nodegraphics"/>
  <key attr.name="url" attr.type="string" for="edge" id="d4"/>
  <key attr.name="description" attr.type="string" for="edge" id="d5"/>
  <key for="edge" id="d6" yfiles.type="edgegraphics"/>
  <graph edgedefault="directed" id="G">
    <node id="n0"><data key="d3"><y:ShapeNode>
        <y:Geometry height="30.0" width="110.0" x="5.6843418860808015E-14"
y="220.0"/>
        <y:Fill color="#CCCCFF" transparent="false"/>
        <y:BorderStyle color="#000000" type="line" width="1.0"/>
        <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="12" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
height="18.701171875" modelName="internal" modelPosition="c" textColor="#000000"
visible="true" width="86.7109375" x="11.64453125" y="5.6494140625">藝術
家:644.471</y:NodeLabel><y:Shape type="rectangle"/></y:ShapeNode></data>
    </node>
    <node id="n1"><data key="d3"><y:ShapeNode>
        <y:Geometry height="30.0" width="110.0" x="0.0" y="0.0"/>
        <y:Fill color="#CCCCFF" transparent="false"/>
        <y:BorderStyle color="#000000" type="line" width="1.0"/>
        <y:NodeLabel alignment="center" autoSizePolicy="content" fontFamily="Dialog"
fontSize="12" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
height="18.701171875" modelName="internal" modelPosition="c" textColor="#000000"
visible="true" width="74.7109375" x="17.64453125" y="5.6494140625">藝
術:363.933</y:NodeLabel><y:Shape type="rectangle"/></y:ShapeNode></data>
    </node>
    <edge id="e0" source="n0" target="n1"><data key="d6">
      <y:PolyLineEdge>
        <y:Path sx="0.0" sy="0.0" tx="0.0" ty="0.0"/>
        <y:LineStyle color="#000000" type="line" width="1.0"/>
        <y:Arrows source="none" target="standard"/>
        <y:EdgeLabel alignment="center" distance="2.0" fontFamily="Dialog" font-
Size="12" fontStyle="plain" hasBackgroundColor="false" hasLineColor="false"
height="18.701171875" modelName="six_pos" modelPosition="tail" preferredPlace-
ment="anywhere" ratio="0.5" textColor="#000000" visible="true" width="47.376953125"
x="2.0000000000000284" y="-104.3505859375">839.079</y:EdgeLabel>
        <y:BendStyle smoothed="false"/></y:PolyLineEdge></data>
    </edge>
  </graph>
  <data key="d0">
    <y:Resources/>
  </data>
</graphml>
```

Fig. 7.8 Sample of a generated GraphML document of *DOG*

7.3.2 Experimental Results

Number of nodes generated

Figure 7.9 summarizes the outcomes of the experiment about the number of nodes generated in the domain ontology graph generation process (of the domain "文藝 Arts and Entertainments") with different threshold values. The result shows that for all threshold θ_u values, the number of nodes proportionally goes down when threshold θ_v is increased by 10. Therefore we summarized the relation between the thresholds θ_v to the rates of nodes generated in Figure 7.10 and Table 7.2. It shows that different threshold θ_u affects the generated number of nodes with a similar rate.

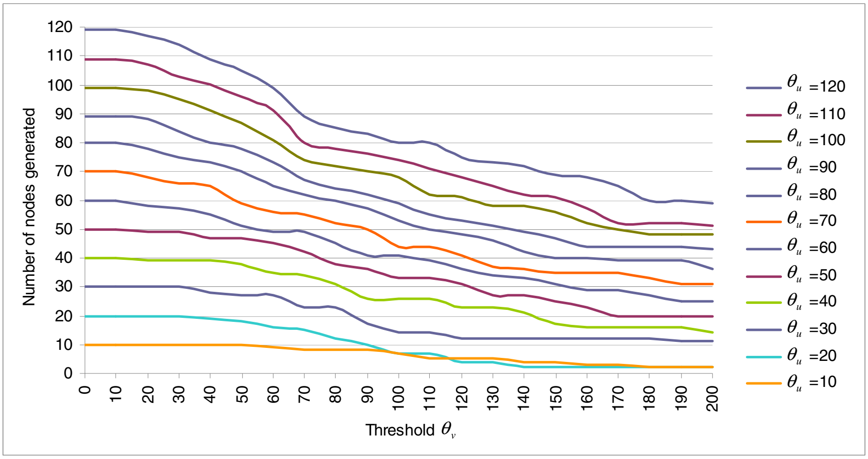


Fig. 7.9 Number of nodes generated for different threshold values

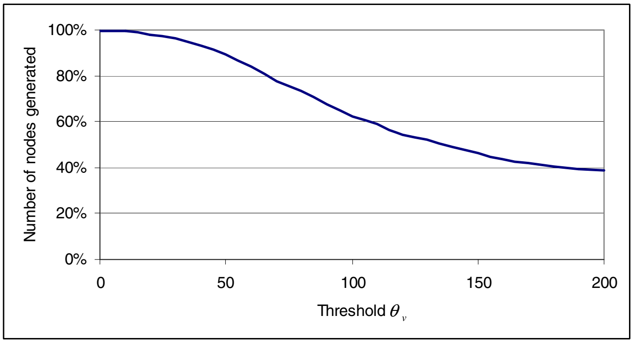


Fig. 7.10 The rates of nodes generated for different threshold values

Table 7.2 Details of the rates of nodes generated for different threshold values

θ_v	Average rate of the generated number of nodes
200	38.66 %
190	39.63 %
180	40.42 %
170	42.28 %
160	43.64 %
150	46.28 %
140	48.69 %
130	51.93 %
120	54.36 %
110	58.95 %
100	62.30 %
90	67.65 %
80	73.28 %
70	77.81 %
60	83.98 %
50	89.52 %
40	93.10 %
30	96.29 %
20	98.11 %
10	99.68 %
0	99.67 %

Number of edges generated

Figure 7.11 summarizes the outcomes of the experiment about the number of edges generated in the *DOG* generation process (of the domain "文藝 Arts an*d Entertainments*") with different threshold values. The result shows that, for all threshold θ_u values, the number of edges exponentially goes down when threshold θ_v is increased by 10. Therefore we summarized the relation between the thresholds θ_v to the rates of nodes generated in Figure 7.12. It shows that different threshold θ_u affects the generated number of edges with a similar exponential rate. The rate of 100% denotes that all edges between every pair of nodes are generated. That means the generated domain ontology graph with 100% rate of edges is a complete graph. If the graph contains *n* number of nodes, the 100% rate of edges is n^2-n.

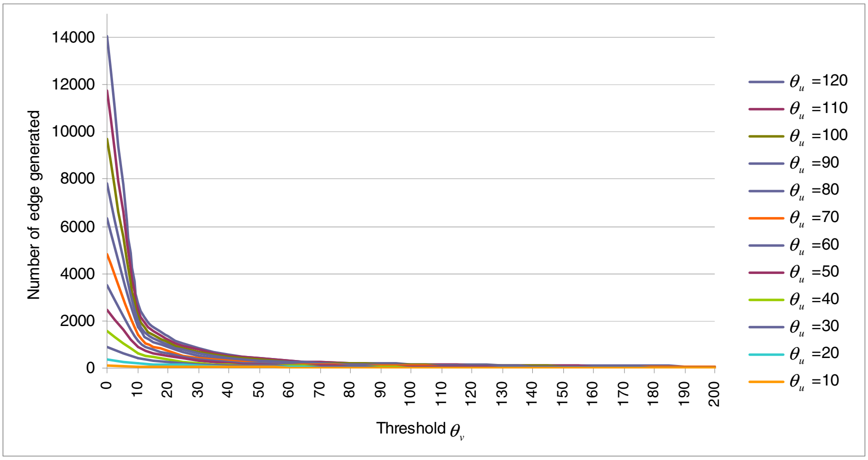


Fig. 7.11 Number of edges generated for different threshold setting

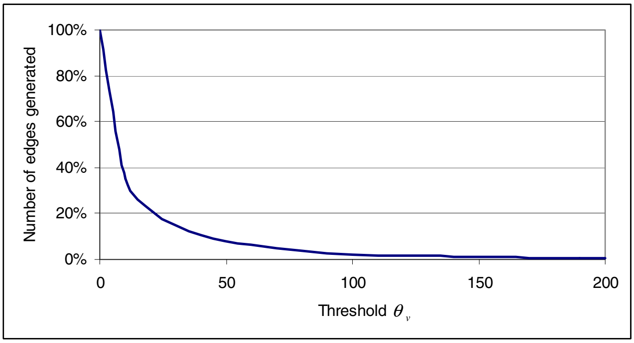


Fig. 7.12 The rate of edges generated for different threshold setting

Example of Generated Domain Ontology Graph with Different Thresholds

Table 7.3 shows the top 20 term-class entries of the domain (class) of “文藝 *Arts and Entertainments*”. In this example, θ_u is set to 20, θ_v is set within the range of 200 to 0. Table 7.4 provides the statistical result of the generated ontology graph about the number of nodes and edges with different θ_v values. The graphical result of the ontology graphs for θ_v = 200, 150, and 100 to 0 are generated in GraphML file format and further visualized with auto circular layout by yED (yED 2009), as shown in Figures 7.13 to 7.33 respectively.

Table 7.3 Top 20 terms of domain "文藝" in ontology learning

Rank	Term	χ^2	R	Rank	Term	χ^2	R
1	藝術 (arts)	1014.18	7.552083	11	戲劇 (opera)	392.7607	9.387755
2	作品 (works)	979.3634	8.992248	12	音樂 (music)	386.0206	6.542056
3	創作 (creative)	975.1136	9.478261	13	節目 (show)	357.0295	7.605634
4	演出 (perform)	748.1122	8.495575	14	舞台 (stage)	349.994	7.846154
5	文藝 (literature)	688.607	8.47619	15	表演 (act)	343.3402	6.595745
6	觀眾 (audience)	666.3134	8.585859	16	美術 (painting)	330.3707	7.051282
7	文化 (culture)	585.201	5.131086	17	風格 (style)	329.2021	8.6
8	藝術家 (artist)	572.6829	9.305556	18	舉辦 (hold)	329.0934	5.125
9	畫 (draw)	512.5542	6.27451	19	劇團 (troupe)	327.732	9.5
10	演員 (actor)	411.4805	9.090909	20	歌舞 (sing)	318.0413	9.069767

Table 7.4 DOC generation result of the domain "文藝 (Arts and Entertainments)"

Domain	θ_u	θ_v	Num of nodes generated	Num of edges generated	Figure
文藝	20	200	2	1	7.13
文藝	20	150	2	1	7.14
文藝	20	100	7	4	7.15
文藝	20	90	10	7	7.16
文藝	20	80	12	11	7.17
文藝	20	70	15	19	7.18
文藝	20	60	16	29	7.19
文藝	20	50	18	43	7.20
文藝	20	40	19	73	7.21
文藝	20	30	20	111	7.22
文藝	20	20	20	148	7.23
文藝	20	10	20	208	7.24
文藝	20	0	20	380	7.25
文藝	120	200	27	70	7.26
文藝	120	160	40	95	7.27
文藝	120	120	54	140	7.28
文藝	120	100	68	177	7.29
文藝	120	80	72	231	7.30
文藝	120	60	108	340	7.31
文藝	120	40	120	580	7.32
文藝	120	20	120	1339	7.33

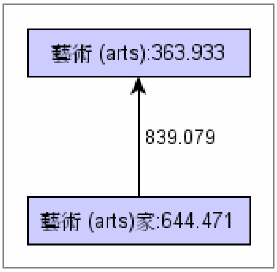


Fig. 7.13 $\theta_u = 20,\ \theta_v = 200$

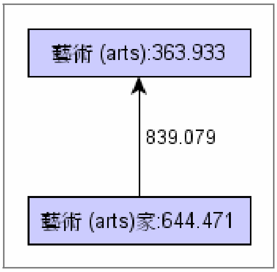


Fig. 7.14 $\theta_u = 20,\ \theta_v = 150$

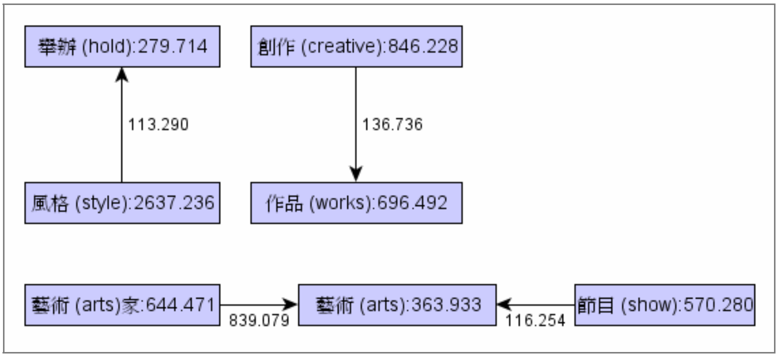


Fig. 7.15 $\theta_u = 20,\ \theta_v = 100$

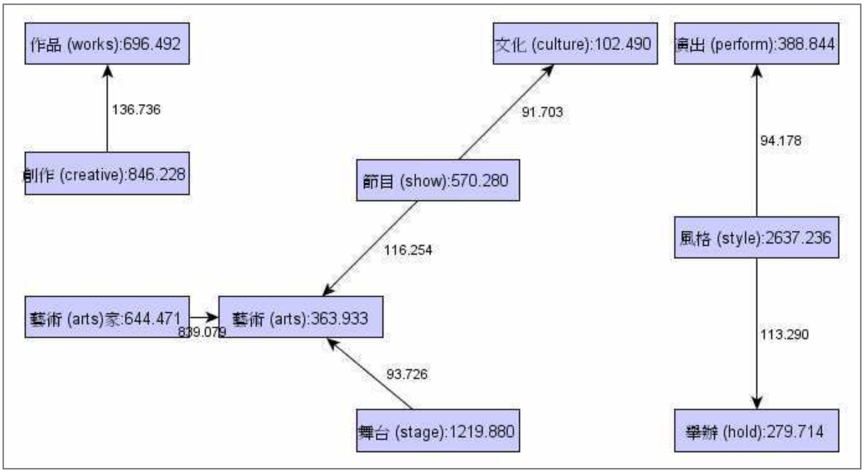


Fig 7.16 $\theta_u = 20,\ \theta_v = 90$

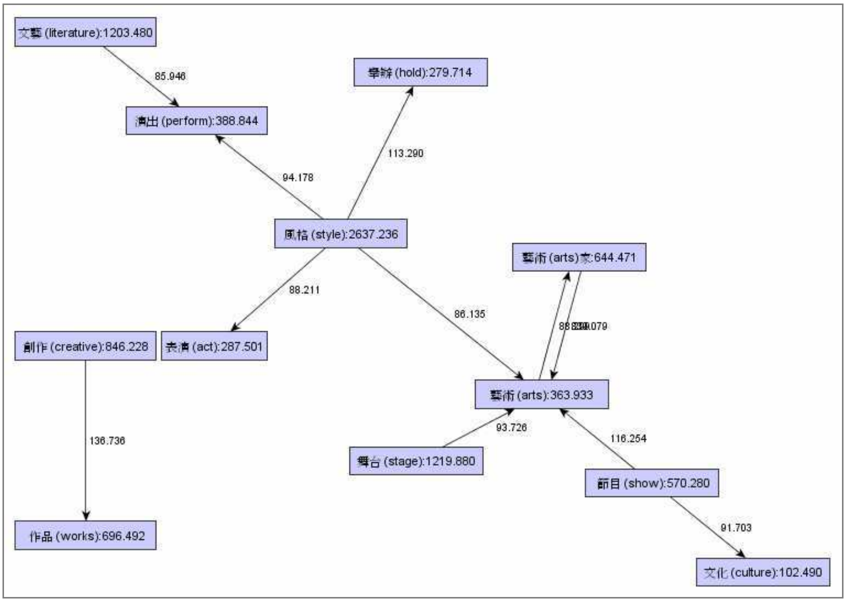


Fig. 7.17 $\theta_u = 20,\ \theta_v = 80$

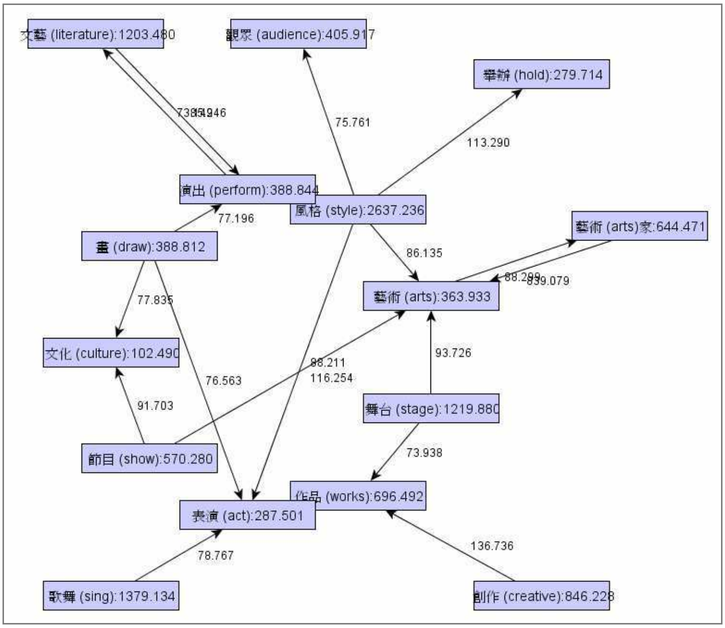


Fig 7.18 $\theta_u = 20,\ \theta_v = 70$

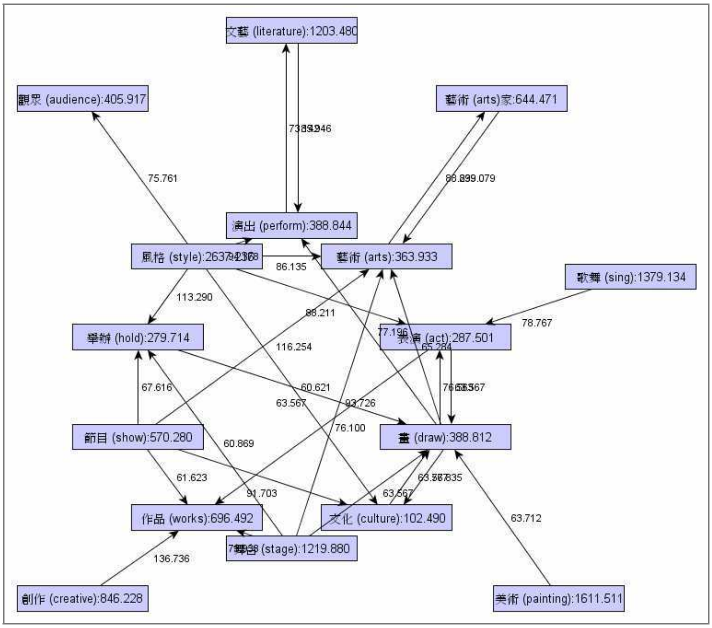


Fig. 7.19 $\theta_u = 20,\ \theta_v = 60$

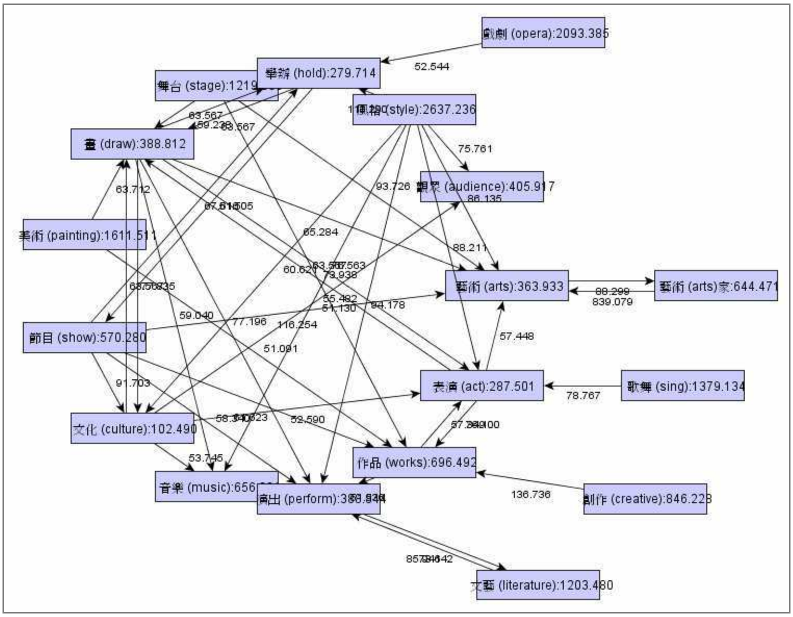


Fig. 7.20 $\theta_u = 20,\ \theta_v = 50$

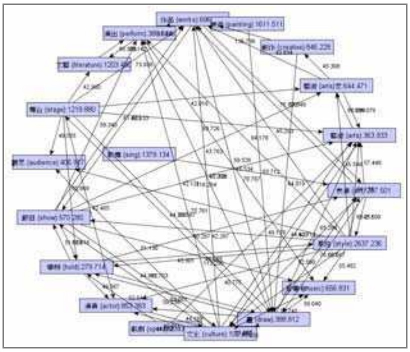


Fig. 7.21 $\theta_u = 20,\ \theta_v = 40$

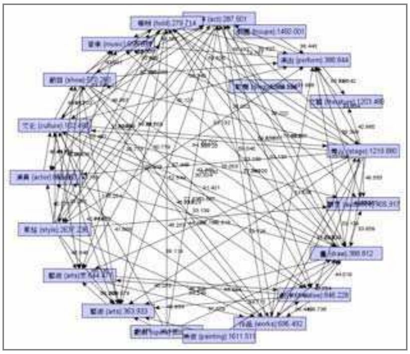


Fig. 7.22 $\theta_u = 20,\ \theta_v = 30$

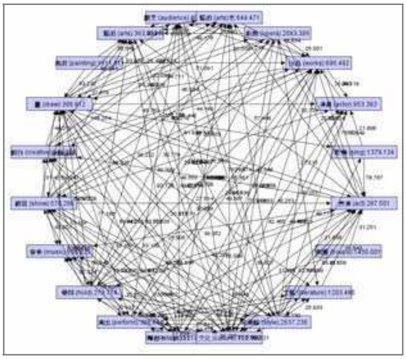


Fig. 7.23 $\theta_u = 20,\ \theta_v = 20$

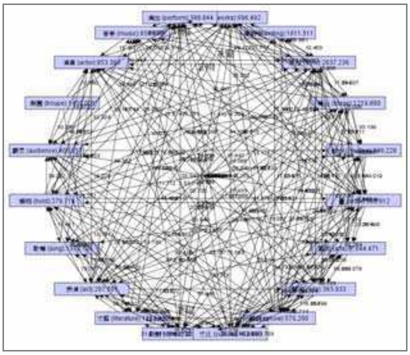


Fig. 7.24 $\theta_u = 20,\ \theta_v = 10$

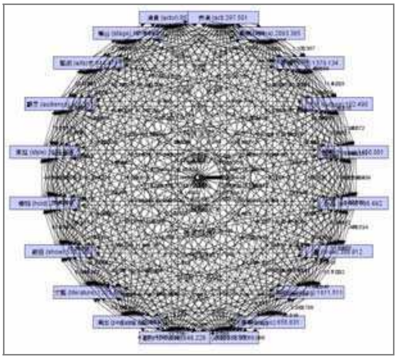


Fig. 7.25 $\theta_u = 20,\ \theta_v = 0$

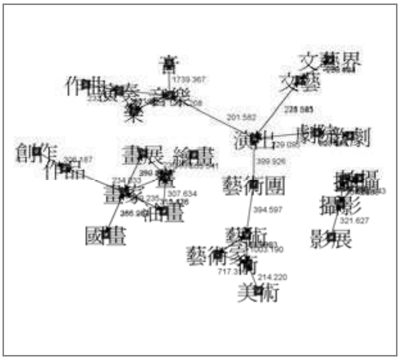


Fig. 7.26 $\theta_u = 120,\ \theta_v = 200$

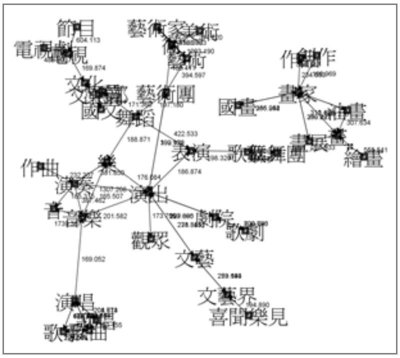


Fig. 7.27 $\theta_u = 120,\ \theta_v = 160$

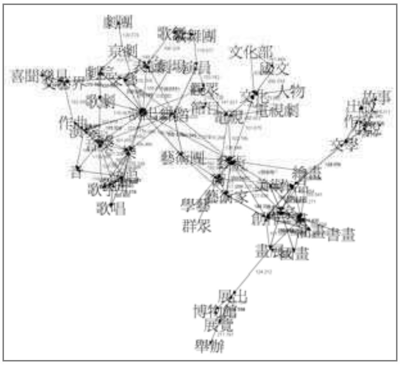


Fig. 7.28 $\theta_u = 120,\ \theta_v = 120$

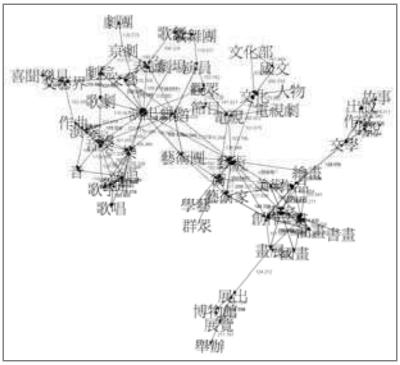


Fig. 7.29 $\theta_u = 120,\ \theta_v = 100$

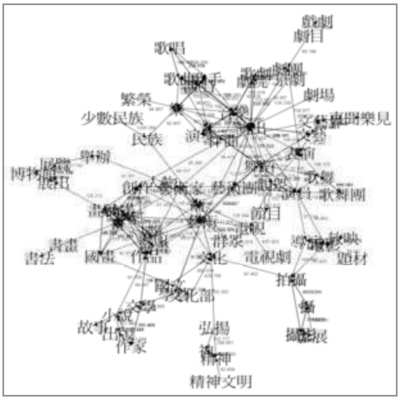


Fig. 7.30 $\theta_u = 120,\ \theta_v = 80$

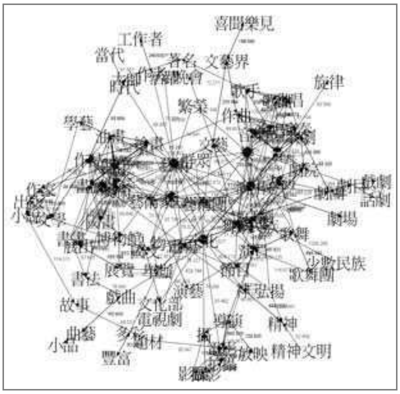


Fig. 7.31 $\theta_u = 120,\ \theta_v = 60$

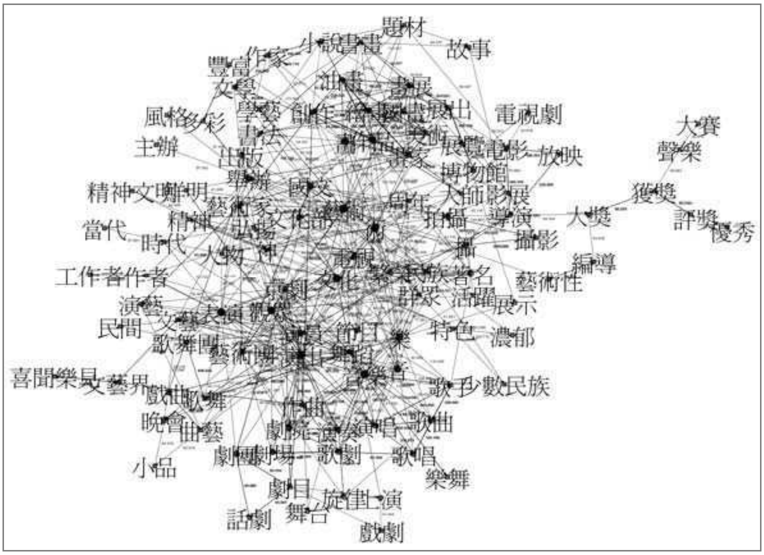


Fig. 7.32 $\theta_u = 120,\ \theta_v = 40$

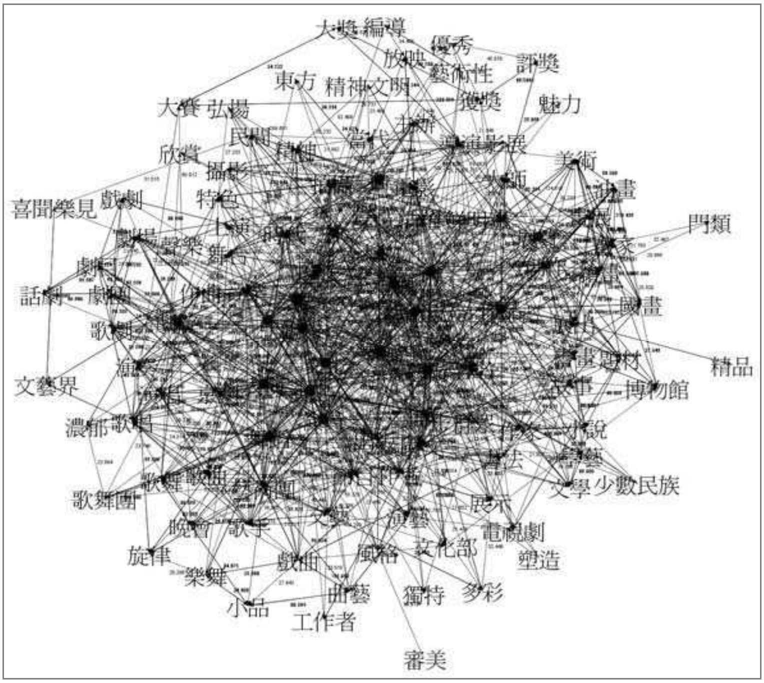


Fig. 7.33 $\theta_u = 120,\ \theta_v = 20$

Chapter 8
Ontology Graph Operations

Abstract. In this chapter, we define different ontological operations (such as similarity measurement and ontology graph based querying) that can be carried out with the use of generated Domain Ontology Graphs. These operations can be applied to develop various ontology based applications such as text classification, search engine, etc. This is the last module of the KnowledgeSeeker system and all modules developed in the KnowledgeSeeker can improve traditional information system with higher efficiency. In particular, it can increase the accuracy of a text classification system, and also enhance the search intelligence in a search engine.

8.1 Ontology Graph Matching and Querying Process

In the previous chapters, we introduced the KnowledgeSeeker system framework, including the ontology modeling, learning and generation modules. The ontology modeling modules provide the format and structure definition of Ontology Graph (*OG*). The ontology learning and generation modules provide the steps of learning and generating a Domain Ontology Graph (*DOG*). The next module in KnowledgeSeeker is the Ontology Graph querying module. It defines different kinds of operation which use *DOG* and the operation methods.

8.2 Introduction to Ontology Matching and Mapping

Ontology matching and mapping process involves two ontologies, and it is aimed to merge two ontologies into one by semantic matching and similarity measurement. The Ontology matching takes two different ontologies and produces mappings between those concepts of the two ontologies. The process requires analyzing the semantic information in the related ontologies. The analysis including semantic mapping and similarity measurement processes is computed automatically, to create and derive a new ontology through the process (Figure 8.1).

E.H.Y. Lim, J.N.K. Liu, and R.S.T. Lee: Knowledge Seeker, ISRL 8, pp. 121 – 142.
springerlink.com

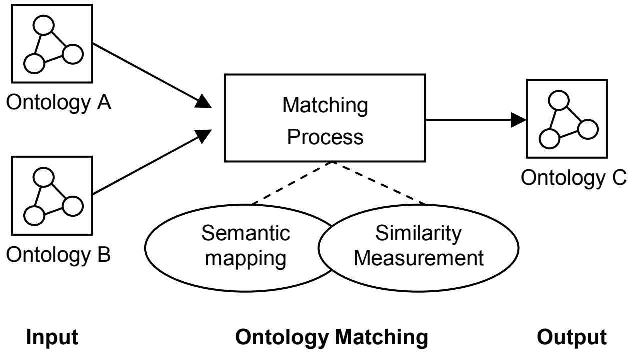


Fig. 8.1 Ontology matching overviews

8.2.1 Ontology Graph Matching Methods

Semantic mapping and similarity measurement play important roles in ontology matching process. The semantic mapping process is to analyze the relationship between two elements, and the similarity measurement process is to compute the distance of those related elements.

8.2.1.1 Semantic Mapping Function

In semantic mapping, we try to find out the terms in two different sources that are related. The related terms do not necessarily to be exactly the same, for examples: we can map the term "*school*" from source S_1 to the term "*college*" in source S_2 as equivalence. Similarly, to map the terms " *teacher*" to "*lecturer*", "*pupil*" to "*student*", and "*class*" to "*lecture*" as equivalence (Figure 8.2). However, for simplification sense, the automatic mapping process of two sources maps two terms as equivalent only when both terms are exactly the same. In addition, we also assign a weight of equivalent between those mapped terms, to express how closely the mapped terms are equivalent to each other.

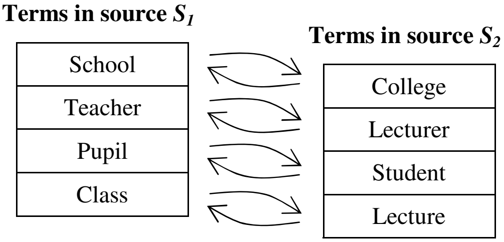


Fig. 8.2 Terms mapping example

The mapping definition is a 3-tuple containing the elements:

$$M =< t_i, t_j, w >$$

Where:

- t_i is the term appearing in source S_1
- t_j is the equivalent term appearing in source S_2
- w is the assigned weight to the term mapping

8.2.1.2 Similarity Measurement Function

In similarity measurement, two different components are taken into comparison. The comparison returns a numerical value indicating how similar of those components are. The similarity function between two components C_1 and C_2 is defined as $sim(C_1,C_2)$ and there several similarity measurement is useful in Ontology Graph matching process.

Equality Similarity

For some component, like the terms in Ontology Graph, two components are defined as equal if both terms are exactly the same:

$$sim_{equality}(C_1,C_2) = \begin{cases} 1 & if\ C_1 = C_2 \\ 0 & else \end{cases}$$

Jacquard Similarity

If two components for comparison are not just single terms, but two different sets of terms, the similarity of these components can be calculated based on the overlapping individuals of the sets:

$$sim_{jacquard}(C_1,C_2) = \frac{|C_1 \cap C_2|}{|C_1 \cup C_2|}$$

Euclidean distance

This measurement is to compare the distance between two components in a vector space model. For example, if the two sets of terms (the two components to be measured) are represented in a vector space model with weight assigned to each term in the sets, this calculation can measure the distance between two vectors. Geometrically, it is the length of the segment joining two components C_1 and C_2:

$$d(C_1,C_2) = \sqrt{\sum_{i=1}^{n}(C_{1_i} - C_{2_i})^{2}}$$

Cosine similarity

Cosine similarity is used to calculate the similarity between two vectors of n dimensions by measuring the cosine of angle between them. If the two components

C_1 and C_2 are represented as two vectors, the comparison measure is the dot product of C_1 and C_2 and further divided by the Euclidean distance between C_1 and C_2,.

$$sim_{\cos ine}(C_1, C_2) = \frac{C_1 \bullet C_2}{|C_1||C_2|}$$

8.3 Matching Different Components with Ontology Graph

The general process of Ontology Graph matching requires two components to be provided: the first one is a main knowledge which is in the form of Ontology Graph and the second one is a source input which is in the form of any components in the form of Ontology Graph / Document (text) / or set of terms. The third one is the final mapping result which is in the form of Ontology Graph. The target result is a new Ontology Graph derived from the mapping process. It is regarded as a derived additional knowledge through the matching process of two provided components. Both source input and target output contain semantic matching to the core knowledge, meaning that the matching process from source knowledge to target knowledge relies on the semantic mapping between both of them and the provided Ontology Graph, as shown in Figure 8.3.

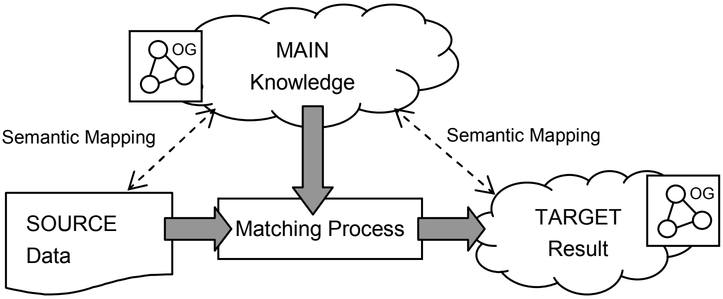


Fig. 8.3 Components of Ontology Graph mapping process

Three Components in Ontology Graph Mapping

- SOURCE Data (*Input Component 1*) – a source of data which is provided as an input in the matching process. It is aimed to match into the MAIN knowledge (an existing Ontology Graph) and obtain more knowledge about the source data.
- MAIN Knowledge (*Input Component 2*) – a provided knowledge which is an existing Ontology Graph obtained from ontology learning and generation process, such as a Domain Ontology Graph.
- TARGET Result (*Out Component*) – a target result of the knowledge obtained through the matching process. The target knowledge is a sub-graph of the mapped Ontology Graph and contains mapping and relations to it.

8.3.1 Matching Terms to Domain Ontology Graph

Concept Formation

Matching a single or multiple terms to a domain Ontology Graph is aimed to extracting more knowledge about the term. In Ontology Graph, the definition of "*term*" is different from that of "*concept*" – a term is only a lexical symbol that represents an entry of a node, while a "*concept"* is formulated by multiple nodes with relations. So we defined "*concept*" as a term that has relations to other terms. This concept formation is done by matching terms to a domain Ontology Graph, as to extract the knowledge of the term about a certain domain.

Concept Formation Components

- SOURCE Data: Single or multiple term(s)
- MAIN Knowledge: A Domain Ontology Graph (*DOG*)
- TARGET Result: An Ontology Graph that describes the term(s)

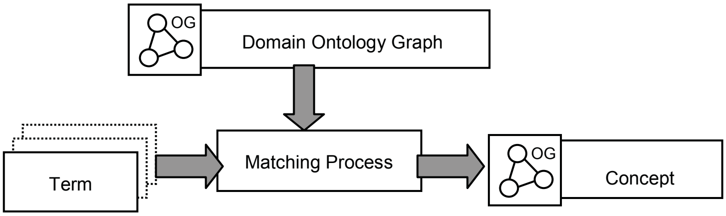


Fig. 8.4 Components of concept formation process

Concept Formation Process Description

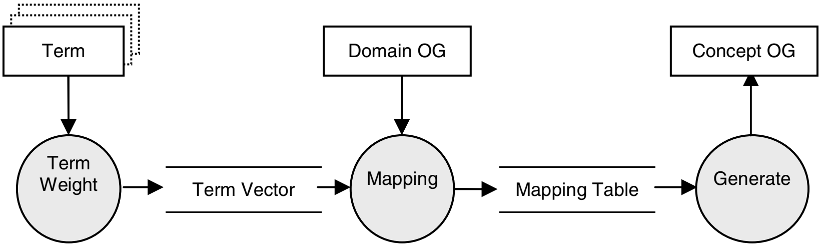


Fig. 8.5 Process of concept formation

Input: An input term list T_S containing at least one term and a *DOG* OG_D.
Output: A concept *OG* OG_C representing the input term(s).

Example 8.1 – Domain Ontology Graph OG_A Definition

For simplification, a sample of Domain Ontology Graph (*DOG*) OG_A which contains only terms and relations representing the domain *A* is defined as follows. The tabular form of OG_A is shown in Table 8.1, and graphical representation of OG_A is shown in Figure 8.6.

- $T{:}OG_A = \{A, B, C, D, E\}$
- $R{:}OG_A = \{(A, A, 1), (A, B, 0.1), (A, C, 0.2), (A, D, 0.3), (A, E, 0.4), (B, A, 0.1), (B, B, 1), (B, C, 0.2), (B, D, 0.3), (B, E, 0.4), (C, A, 0.1), (C, B, 0.2), (C, C, 1), (C, D, 0.3), (C, E, 0.4), (D, A, 0.1), (D, B, 0.2), (D, C, 0.3), (D, D, 1), (D, E, 0.4), (E, A, 0.1), (E, B, 0.2), (E, C, 0.3), (E, D, 0.4), (E, E, 1)\}$

Table 8.1 Table of the terms and relations in OG_A

	A	*B*	*C*	*D*	*E*
A	*1*	*0.1*	*0.2*	*0.3*	*0.4*
B	*0.1*	*1*	*0.2*	*0.3*	*0.4*
C	*0.1*	*0.2*	*1*	*0.3*	*0.4*
D	*0.1*	*0.2*	*0.3*	*1*	*0.4*
E	*0.1*	*0.2*	*0.3*	*0.4*	*1*

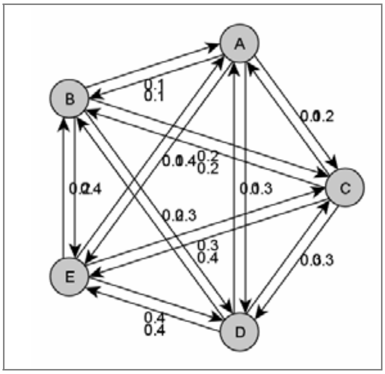


Fig. 8.6 Ontology Graph OG_A

Example 8.2 – Concept Formation

A "*concept*" is a large knowledge object that based on a single term or multiple terms with relations in the Ontology Graph OG_A. The most basic concept is defined by using a single term *t*, where $t \in T{:}OG_A$. Therefore all single-term concepts in OG_A can be formulated including: c_A for *concept*(*"A"*), c_B for *concept*(*"B"*), c_C for *concept*(*"C"*), c_D for *concept*(*"D"*), c_E for *concept*(*"E"*). The steps of formulating concepts are shown below and the visualized *OG* for concept c_A, c_B, c_C, c_D, c_E are shown in Figures 8.7 to 8.11 correspondingly:

Step 1: Obtain the term list from *concept(t)* as T_S
Step 2: Obtain the relation set R_C from $R{:}OG_A$ where at least one term is in T_S
Step 3: Obtain all distinct terms as term list from the relation set
Step 4: Match the new term list for concept (t) as $T_C = T_S \cup T_A$
Step 5: Create the new *OG* with T_C as term list and R_C

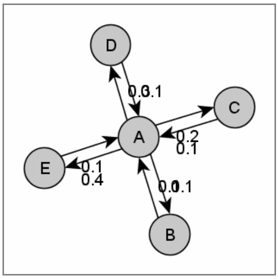


Fig. 8.7 *OG* of Concept c_A

Table 8.2 Mapping of concept c_A

	A	*B*	*C*	*D*	*E*
A	*1*	*0.1*	*0.2*	*0.3*	*0.4*
B	*0.1*	-	-	-	-
C	*0.1*	-	-	-	-
D	*0.1*	-	-	-	-
E	*0.1*	-	-	-	-

8.8

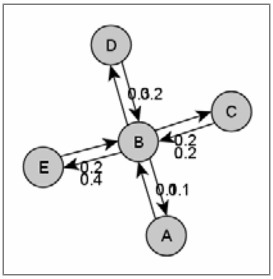


Fig. 8.8 *OG* of Concept c_B

Table 8.3 Mapping of concept c_B

	A	*B*	*C*	*D*	*E*
A	-	*0.1*	-	-	-
B	*0.1*	*1*	*0.2*	*0.3*	*0.4*
C	-	*0.2*	-	-	-
D	-	*0.2*	-	-	-
E	-	*0.2*	-	-	-

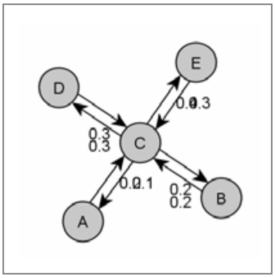


Fig. 8.9 *OG* of Concept c_C

Table 8.4 Mapping of concept c_C

	A	*B*	*C*	*D*	*E*
A	-	-	*0.2*	-	-
B	*0.1*	*1*	*0.2*	*0.3*	*0.4*
C	-	-	*1*	-	-
D	-	-	*0.3*	-	-
E	-	-	*0.3*	-	-

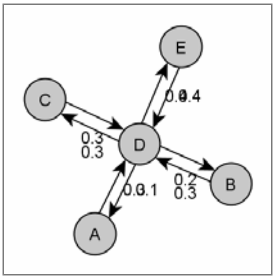


Fig. 8.10 *OG* of Concept c_D

Table 8.5 Mapping of concept c_D

	A	*B*	*C*	*D*	*E*
A	-	-	-	*0.3*	-
B	-	-	-	*0.3*	-
C	-	-	-	*0.3*	-
D	*0.2*	*0.2*	*0.3*	*1*	*0.4*
E	-	-	-	*0.4*	-

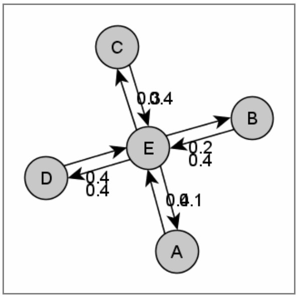


Fig. 8.11 *OG* of Concept c_E

Table 8.6 Mapping of concept c_E

	A	*B*	*C*	*D*	*E*
A	-	-	-	-	*0.4*
B	-	-	-	-	*0.4*
C	-	-	-	-	*0.4*
D	-	-	-	-	*0.4*
E	*0.1*	*0.2*	*0.3*	*0.4*	*1*

8.3.2 Matching Text Document to Domain Ontology Graph

Document Ontology Graph (DocOG) Generation

A Document Ontology Graph (*DocOG*) is a type of Ontology Graph that used to represent the content of a text document. Traditional information system usually represents documents by term vectors. In KnowledgeSeeker system, we proposed to use the Ontology Graph to represent the content about a text document. In addition, the *DocOG* can also describe more information about the document, such as the related knowledge of a certain domain. This can be done by matching the text document to a Domain Ontology Graph (*DOG*) to acquire more knowledge about the related domain.

The matching of a text document to a *DOG* aims at extracting more knowledge about the domain inside the document. Text document is often represented by a

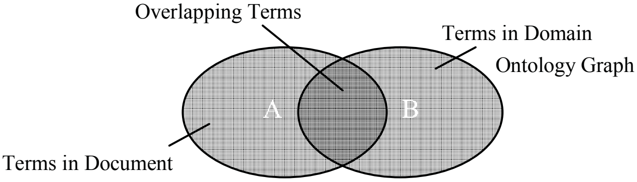


Fig. 8.12 Mapping overlapping terms in document and Domain Ontology Graph

list of terms (a weighted term vector). We match a text document to a *DOG* to create mappings between them if they have an intersection of same terms (Figure 8.12). This process can relate a document to a particular domain. The process can be used to extract more knowledge about the document and also measure the similarity of the document to the matched target *DOG*.

Document Ontology Graph Formation Components

- SOURCE Data: A document written in the form of text
- MAIN Knowledge: A Domain Ontology Graph (*DOG*) of a certain domain
- TARGET Result: A Document Ontology Graph (*DocOG*) describing the document

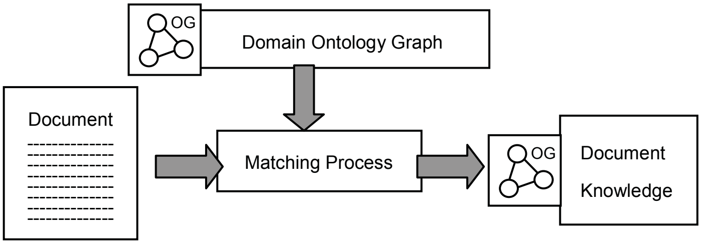


Fig. 8.13 Components of Document Ontology Graph generation process

Document Ontology Graph Generation Process Description

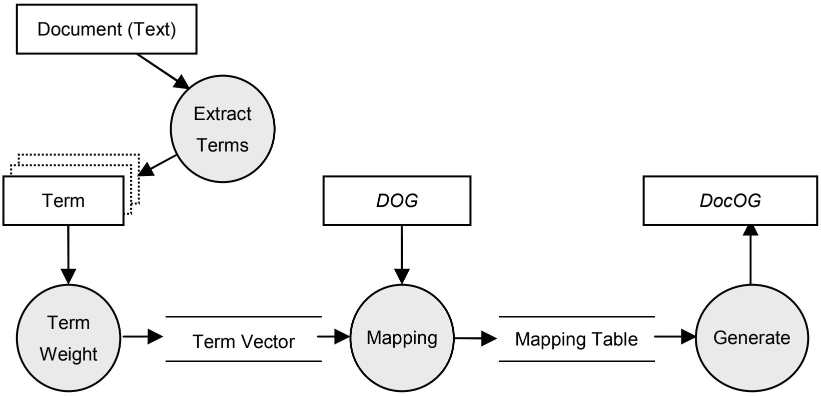


Fig. 8.14 Process of Document Ontology Graph (*DocOG*) extraction

Input: An input document (text) and a Domain Ontology Graph (*DOG*) – OG_d for domain *d*.
Output: A Document Ontology Graph (*DocOG*) – OG_{doc} representing the input document *doc*.

Example 8.3 – Document Ontology Graph Extraction

The Domain Ontology Graph OG_A definition is referenced from Example 8.1. Two examples about documents d_1 and d_2 which are defined as follows to illustrate the generation of their corresponding Document Ontology Graph (OG_{d_1} and OG_{d_2}):

Step 1: Obtain the document content

- *d_1: A–A–B–D* (Document-length = 4).
- *d_2: D–D–D–E* (Document length = 4).

Step 2: Transformed to weighted term vector

The weight of every term in each document is weighted by W_{t_i,d_j} where t_i represents the i^{th} distinct term in the document *j*. W_{t_i,d_j} is defined as:

$$W_{t_i,d_j} = \frac{tf_{i,j}}{dl_j}$$

- *tf* denotes the frequency of term *i* appearing in document *j*
- *dl* denotes the document length, i.e. the size of the term list of document *j*

The transformed term vectors of the two documents are as follows:

- T_{d_1} = *{(A, 0.5), (B, 0.25), (D, 0.25)}* (Num-of-term = 3).
- T_{d_2} = *{(D, 0.75), (E, 0.25)}* (Num-of-term = 2).

Step 3: Term List creation for two *DocOGs* OG_{d_1} and OG_{d_2} :

- $T:OG_{d_1}$ = *{A, B, D}*
- $T:OG_{d_2}$ = *{D, E}*

Step 4: Concept Formation. The results are shown in Tables 8.7 to 8.11.

- $OG_{d_1} : c_{A,d_1}, c_{B,d_1}, c_{D,d_1}$
- $OG_{d_2} : c_{D,d_2}, c_{E,d_2}$

Table 8.7 c_{A,d_1}

	A	*B*	*C*	*D*	*E*
A	*0.5*	*0.05*	*0.1*	*0.15*	*0.2*
B	*0.05*	-	-	-	-
C	*0.05*	-	-	-	-
D	*0.05*	-	-	-	-
E	*0.05*	-	-	-	-

Table 8.8 c_{B,d_1}

	A	*B*	*C*	*D*	*E*
A	-	*0.025*	-	-	-
B	*0.025*	*0.25*	*0.05*	*0.075*	*0.1*
C	-	*0.05*	-	-	-
D	-	*0.05*	-	-	-
E	-	*0.05*	-	-	-

Table 8.9 c_{D,d_1}

	A	*B*	*C*	*D*	*E*
A	-	-	-	*0.075*	-
B	-	-	-	*0.075*	-
C	-	-	-	*0.075*	-
D	*0.025*	*0.05*	*0.075*	*0.25*	*0.1*
E	-	-	-	*0.1*	-

Table 8.10 c_{D,d_2}

	A	*B*	*C*	*D*	*E*
A	-	-	-	*0.225*	-
B	-	-	-	*0.225*	-
C	-	-	-	*0.225*	-
D	*0.075*	*0.15*	*0.225*	*0.75*	*0.3*
E	-	-	-	*0.3*	-

Table 8.11 c_{E,d_2}

	A	*B*	*C*	*D*	*E*
A	-	-	-	-	*0.1*
B	-	-	-	-	*0.1*
C	-	-	-	-	*0.1*
D	-	-	-	-	*0.1*
E	*0.025*	*0.05*	*0.075*	*0.1*	*0.25*

Step 5: Ontology graph mapping from the related concepts:

Matching the concepts of the *DocOG* for documents d_1 and d_2:

$$OG_{d_1} = c_{A,d_1} \times c_{B,d_1} \times c_{D,d_1}, \quad OG_{d_2} = c_{D,d_2} \times c_{E,d_2}$$

If the relation of terms t_i and t_j exists more than once among all the formulated concepts, the max weighting for that relation is assigned, i.e. for every t_i and t_j relation, $Rel_S(t_i, t_j, w_{t_i,t_j})$ is selected for $MAX(w_{t_i,t_j})$.

Step 6: Relation set creation for the *DocOG* OG_{d_1} and OG_{d_2} :

- *R:* OG_{d_1} *= {(A, A, 1), (A, B, 0.1), (A, C, 0.2), (A, D, 0.3), (A, E, 0.4), (B, A, 0.1), (B, B, 0.5), (B, C, 0.1), (B, D, 0.15), (B, E, 0.2), (C, A, 0.1), (C, B, 0.1),, (C, D, 0.15),(D, A, 0.1), (D, B, 0.1), (D, C, 0.15), (D, D, 0.5), (D, E, 0.2), (E, A, 0.1), (E, B, 0.1), (E, D, 0.2)}*
- *R:* OG_{d_2} *= {(A, D, 0.3), (A, E, 0.132), (B, D, 0.3), (B, E, 0.132), (C, D, 0.3), (C, E, 0.132), (D, A, 0.1), (D, B, 0.2), (D, C, 0.3), (D, D, 1), (D, E, 0.4), (E, A, 0.033), (E, B, 0.066), (E, C, 0.099), (E, D, 0.3), (E, E, 0*__*.33)}*__

Table 8.12 Terms and relations in OG_{d_1}

	A	*B*	*C*	*D*	*E*
A	*0.5*	*0.05*	*0.1*	*0.15*	*0.2*
B	*0.05*	*0.25*	*0.05*	*0.075*	*0.1*
C	*0.05*	*0.05*	-	*0.075*	-
D	*0.05*	*0.05*	*0.075*	*0.25*	*0.1*
E	*0.05*	*0.05*	-	*0.1*	-

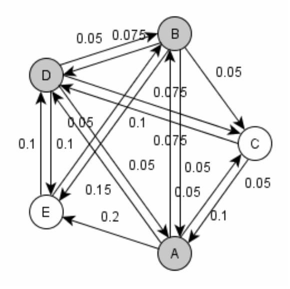


Fig. 8.15 *DocOG* for $d_1 - OG_{d_1}$

Table 8.13 Terms and relations in OG_{d_2}

	A	B	C	D	E
A	-	-	-	*0.225*	*0.1*
B	-	-	-	*0.225*	*0.1*
C	-	-	-	*0.225*	*0.1*
D	*0.075*	*0.15*	*0.225*	*0.75*	*0.3*
E	*0.025*	*0.05*	*0.075*	*0.3*	*0.25*

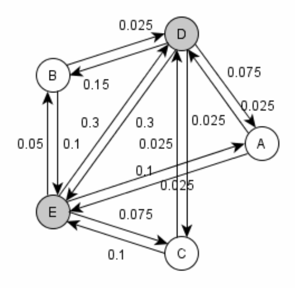


Fig. 8.16 *DocOG* for $d_2 - OG_{d_2}$

8.3.3 Ontology Graph Based Similarity Measurement

Document and domain knowledge comparison

Matching a *DocOG* (for a document) with a *DOG* (for a domain) is aimed to measure the similarity between a document and a particular domain. This is to find out how the document is related to the domain of interest. This matching process is useful in text classification process. When a document is compared to several *DOGs*, the highest ranked *DOG* in the result is the domain that the document is mostly related to.

Document and domain comparison components

- SOURCE Data: A Document Ontology Graph (*DOG*)
- MAIN Knowledge: A Domain Ontology Graph (*DocOg*)
- TARGET Result: A score value representing the similarity

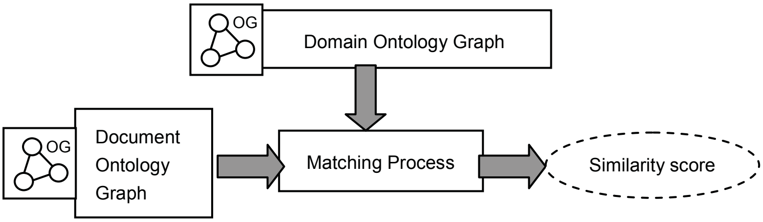


Fig. 8.17 Components of Document and Domain Ontology Graph comparison

Document and Domain Comparison Process Description

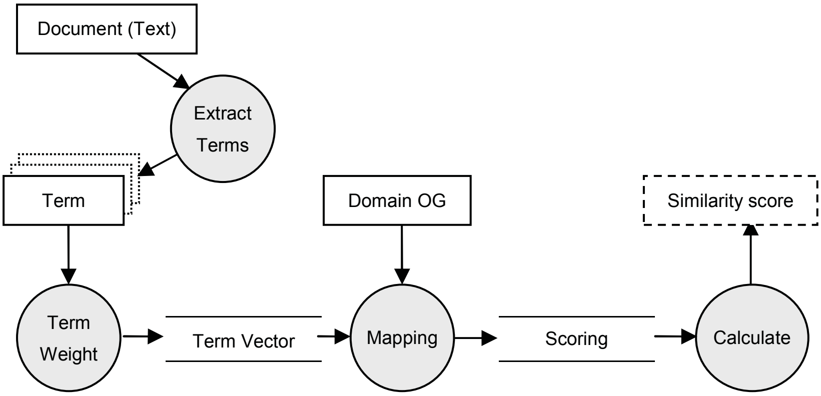


Fig. 8.18 Process of Document and Domain Ontology Graph comparison

Input: A Document Ontology Graph OG_d representing the input document
Output: A similarity score representing the comparison result

Example 8.4 – Document and Domain Ontology Graph Comparison

The domain OG_A definition and the content of two documents d_1 and d_2 used in this example are referenced from Example 8.1. The formation of their corresponding *DocOG* (OG_{d_1} and OG_{d_2}) are also given in that example (refer to Example 8.1). In this example, we illustrate the process of comparing both *DocOGs* to the *DOG*, i.e. comparing OG_{d_1} to OG_A and OG_{d_2} to OG_A. This comparison requires several sub-process including the term matching, semantic mapping, etc. The main step in this comparison process is the similarity measurement method, which is described in the following.

Step 1: Obtain the Domain Ontology Graph OG_A

- OG_A – refer to example the definition in Example 8.1

Step 2: Obtain the Document Ontology Graph by matching to the domain OG_A

- OG_{d_1} (see Table 8.14)
- OG_{d_2} (see Table 8.15)

Table 8.14 Terms and relations in OG_{d_1}

	A	*B*	*C*	*D*	*E*
A	*0.5*	*0.05*	*0.1*	*0.15*	*0.2*
B	*0.05*	*0.25*	*0.05*	*0.075*	*0.1*
C	*0.05*	*0.05*	-	*0.075*	-
D	*0.05*	*0.05*	*0.075*	*0.25*	*0.1*
E	*0.05*	*0.05*	-	*0.1*	-

Table 8.15 Terms and relations in OG_{d_2}

	A	*B*	*C*	*D*	*E*
A	-	-	-	*0.225*	*0.1*
B	-	-	-	*0.225*	*0.1*
C	-	-	-	*0.225*	*0.1*
D	*0.075*	*0.15*	*0.225*	*0.75*	*0.3*
E	*0.025*	*0.05*	*0.075*	*0.3*	*0.25*

Step 3: Obtain the score of each *DocOG* by summing up all the relations, excluding all weight of self-relations (the weight of the term itself)

- $score(OG_{d_1}, OG_A) = 0.05 + 0.1 + 0.15 + 0.2 + 0.05 + 0.05 + 0.075 + 0.1 + 0.05 + 0.05 + 0.075 + 0.05 + 0.05 + 0.075 + 0.1 + 0.05 + 0.05 + 0.1 = 1.425$
- $score(OG_{d_2}, OG_A) = 0.225 + 0.1 + 0.225 + 0.1 + 0.225 + 0.1 + 0.075 + 0.15 + 0.225 + 0.3 + 0.025 + 0.05 + 0.075 + 0.4 = 2.275$

Step 4: Finalizing the similarity scores:

- $sim(OG_{d_1}, OG_A) = score(OG_{d_1}, OG_A) / score\ (OG_A) = 1.425 / 5 = 0.285$
- $sim(OG_{d_2}, OG_A) = score(OG_{d_2}, OG_A) / score\ (OG_A) = 2.275 / 5 = 0.455$

8.3.4 Matching Two Document Ontology Graphs

Comparison between two documents

Matching two document ontology graphs is aimed to measure the similarity between two documents. This is to find out how documents are related to each other. This matching process is useful in text clustering process since it can relate and group highly related documents into cluster while separating unrelated documents from other clusters. It is also useful in some information system such as searching related documents, as to retrieve additional related information about the current document. When two document ontology graphs are compared together, a score value is calculated to represent how close the documents are related.

Documents comparison components

- SOURCE Data: A Document Ontology Graph
- MAIN Knowledge: Another Document Ontology Graph
- TARGET Result: A score value representing the similarity

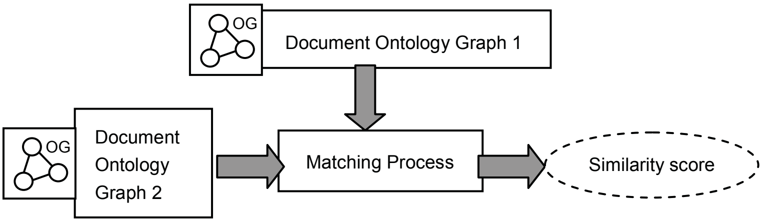


Fig. 8.19 Components of comparison between two documents

Document Comparison Process Description

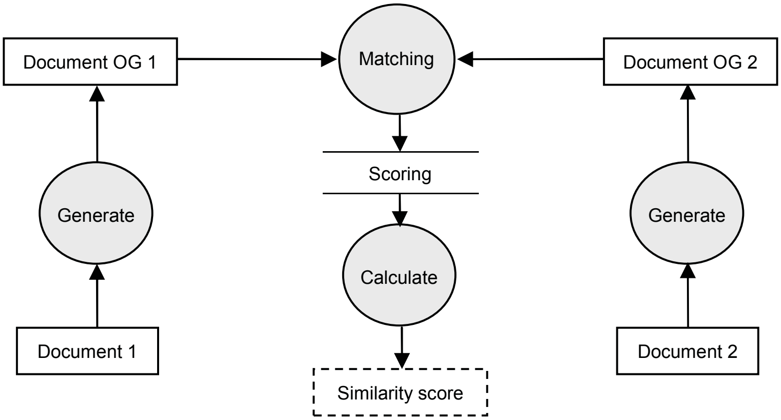


Fig. 8.20 Process of documents comparisons

Input: Two Document Ontology Graphs transformed from two documents (texts)
Output: A similarity score representing the comparison result

Example 8.5 – Comparison between two Document Ontology Graphs

The content of two document d_1 and d_2 used in this example are referenced from Example 8.3. The formation of their corresponding document ontology graph (OG_{d_1} and OG_{d_2}) are also given in the example (Figures 8.15 and 8.16). In this example, we illustrate the process of the comparison between the two documents ontology graph. i.e. comparing OG_{d_1} to OG_{d_2}. This includes the term matching and semantic mapping process. The main step in this comparison process is similar to that of comparing a document ontology graph to a domain ontology graph. This comparison step also requires a similarity measurement, and the result of the comparison gives a similarity score representing how two documents are closely related or unrelated.

Step 1: Obtain the document ontology graphs by matching to the domain OG_A

- OG_{d_1} (see Table 8.13 and Figure 8.15)
- OG_{d_2} (see Table 8.14 and Figure 8.16)

Step 2: Ontology graph matching from all the related concepts

This step matches all intersect concepts between OG_{d_1} and OG_{d_2} to formulate a new ontology graph:

$$OG_m = OG_{d_1} \times OG_{d_2} \,.$$

The similarity measurement between OG_{d_1} and OG_{d_2} is defined as:

$$sim\ (OG_{d_1}, OG_{d_2}) = \frac{OG_{d_1} \cap OG_{d_2}}{OG_{d_1} \cup OG_{d_2}}$$

The minimum weight values for all relations between two term t_i and t_j is assigned for the new ontology graph, i.e. for every t_i and t_j relation, $Rel_S(t_i, t_j, w_{t_i,t_j})$ is selected for $MIN(w_{t_i,t_j})$. Therefore, the relations in the new formulated ontology graph are as follows:

- $R{:}OG_m = R{:}OG_{d_1} \cap R{:}OG_{d_2}$
- *R:OG_m* = *{(A, D, 0.15), (A, E, 0.1), (B, D, 0.075), (B, E, 0.1}, (C, D, 0.075), (D, A, 0.05), (D, B, 0.05), (D, C, 0.075), (D, D, 0.25), (D, E, 0.1), (E, A, 0.025), (E, B, 0.05), (D, E, 0..1)*
- OG_m (see Table 8.17)

Table 8.16 Mapping result

	A	*B*	*C*	*D*	*E*
A	-	-	-	*0.15*	*0.1*
B	-	-	-	*0.075*	*0.1*
C	-	-	-	*0.075*	-
D	*0.05*	*0.05*	*0.075*	*0.25*	*0.1*
E	*0.025*	*0.05*	-	*0.1*	-

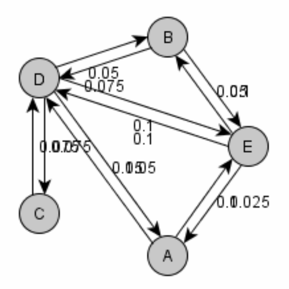


Fig. 8.21 OG of mapping result

Step 3: Obtain the score of the newly formulated ontology graph by summing up all the relation weights, including all weight of self-relations (the weight of the term itself)

- $score\ (OG_m) = 0.15 + 0.1 + 0.075 + 0.1 + 0.075 + 0.05 + 0.05 + 0.075 + 0.25 + 0.1 + 0.025 + 0.05 + 0.1 = 1.2$

Step 4: Finalizing the similarity scores

- Similarity of OG_{d_1} to OG_{d_2} = score (OG_m) / score (OG_{d_2}) = 1.2 / *2.275* = *0.527*
- Similarity of OG_{d_2} to OG_{d_1} = score (OG_m) / score (OG_{d_1}) = 1.2 / *1.425* = *0.842*

8.4 Overviews of Ontology Graph Based Querying

Ontology Graph based querying involves a query and a set of documents which are represented by Ontology Graph model. It is aimed to retrieve a set of documents that are highly related to the query by ontology matching and similarity measurement. The provided query is processed with matching related concepts to every document, and further calculating the similarity score by comparing the weight of those concepts between the query and documents. The higher similarity score denotes a higher relevancy about a document to the query. After a ranking and sorting process according to the calculated scores, a list of documents are retrieved as the querying result.

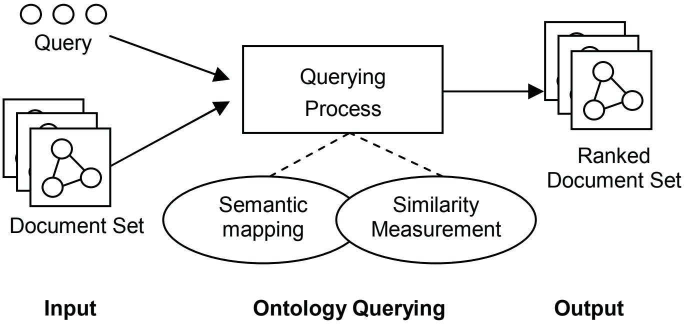


Fig. 8.22 Ontology querying overviews

8.4.1 Ontology Graph Querying Methods

The semantic mapping and similarity measurement process in the ontology graph matching are also used as the similarly measurement in the ontology graph querying process. Before the semantic mapping and similarity measurement are processed, every document is required to be transformed to ontology graph format through the document ontology graph formation process. The semantic mapping process is then used to analyze the relationship between the query and the document ontology graph, and the similarity measurement process is then used to compute the distance between the query and all the transformed document ontology graphs, to provide a ranked documents result.

8.5 Operations in Ontology Graph Based Querying

The operation of Ontology Graph based querying requires two components to be provided: the first one is the query itself, which is provided in the form of a term list (i.e. a list of keywords like the query in traditional search system). The second component is the document to be compared, which are provided in the form of *DocOG*. The *DocOG* of the document is created and generated automatically for a certain domain by the Document Ontology Graph generation process. The target result is the derived Ontology Graph representing the similarity between the query and document. In the querying process, the input query and the outcome ontology graph contain semantic mapping to the document knowledge, and those mappings reveal how the query and the result is related to the document and thus provide the comparison information for documents ranking.

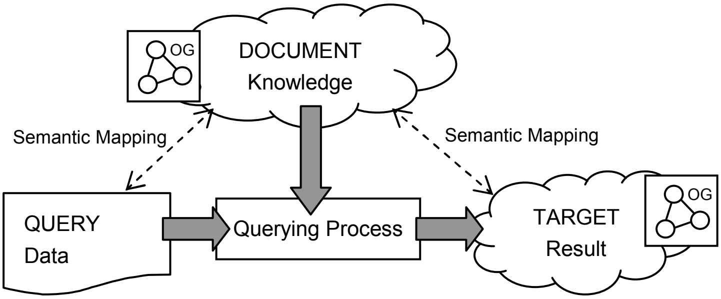


Fig. 8.23 Components of Ontology Graph querying process

Three Components in Ontology Graph Querying

- *QUERY Data (Input Component 1)* – an input of query data which is provided to match and compare with the document knowledge and obtain similarity details.
- *DOCUMENT Knowledge (Input Component 2)* – a provided knowledge in the form of Document Ontology Graph which describes the content of a document, the Document Ontology Graph of that document is generated in the Document Ontology Graph generation process.
- *TARGET Result (Output Component)* – a target result of the knowledge obtained through the querying process. The target knowledge is a comparison sub-graph about the query and the document and contains semantic mapping and relations to them.

8.5.1 Querying Document with Document Ontology Graph

Query and Document Comparison

Matching a term based query to a Document Ontology Graph (*DocOG*) is aimed to compare the knowledge difference and similarity between them. The process mainly matches the same terms in the query and in the *DocOG*, and then by intersecting their relation weight, the similarity score can be calculated.

Query and Document Comparison Components

- QUERY Data: A query provided in a list of terms
- DOCUMENT Knowledge: A *DocOG* describing the content of a document
- TARGET Result: The comparison result of the two inputs

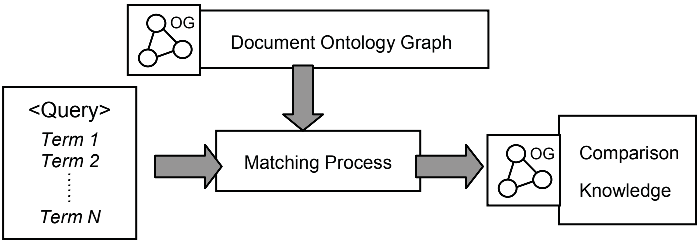


Fig. 8.24 Components of document ontology graph formation process

Document Ontology Graph Formation Process Description

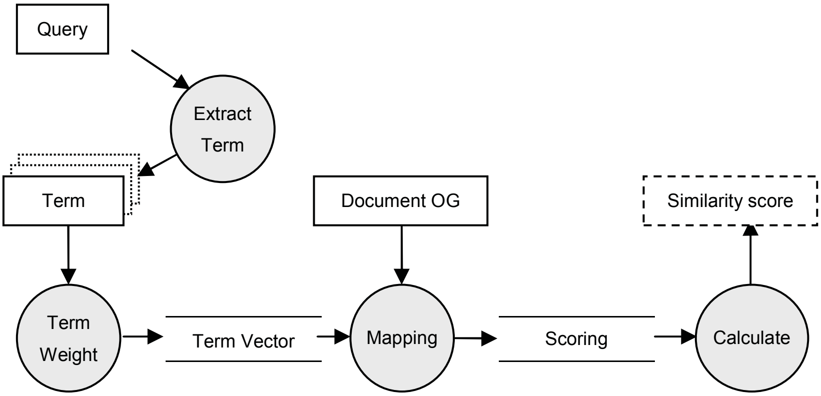


Fig. 8.25 Process of Document Ontology Graph extraction

Input: An input query q and a Document Ontology Graph (*DocOG*) – OG_d
Output: A similarity score representing the comparison result

Example 8.6 – Query and Document Comparison

The content of two documents d_1 and d_2 used in this example are referenced from Example 8.3. The generation results of their corresponding Document Ontology Graphs (OG_{d_1} and OG_{d_2}) were also given in that example. In this example, we illustrate the process of the comparison between a query q and the two Document Ontology Graphs. i.e. comparing q to OG_{d_1} , and q to OG_{d_2}. The comparison steps contain term matching from the query to the *DocOG*, and also the similarity measurement. The comparison result produces a similarity score denoting how the measured documents are related to the query. Therefore, after ranking the document set by the similarity scores, a list of querying results that are sorted by the score are thus produced, where the highest similarity score denotes that the documents are the most relevant to the query.

Step 1: Obtain the query content in the form of a list of terms, 3 queries are provided as examples:

- q_1: *B* (Single term query, query-length = 1)
- q_2: *C* (Single term query, query-length = 1)
- q_3: *B–E* (Multiple term query, query-length = 2)

Step 2: Transformed to weighted term vector for the queries:

The weight of every term in each query is weighted by W_{t_i,q_j} where t_i representing the i^{th} term in the query *j*. W_{t_i,q_j} is defined as:

$$W_{t_i,q_j} = \frac{tf_{i,j}}{|q_j|}$$

- $tf_{i,j}$ denotes the frequency of term *i* appearing the query *j*
- $|q_j|$ denotes the query length, i.e. the number of terms in the query *j*

The transformed term vectors of the three queries are as follows:

- T_{q_1} = *{(B, 1)}* (Num-of-term = 1).
- T_{q_2} = *{(C, 1)}* (Num-of-term = 1).
- T_{q_2} = *{(B, 0.5), (E, 0.5)}* (Num-of-term = 2).

Step 3: Generate the *DocOGs* by matching to the *DOG – OG_A*

- OG_{d_1} (see Table 8.17 and Figure 8.26)
- OG_{d_2} (see Table 8.18 and Figure 8.27)

Table 8.17 Terms and relations in OG_{d_1}

	A	*B*	*C*	*D*	*E*
A	*0.5*	*0.05*	*0.1*	*0.15*	*0.2*
B	*0.05*	*0.25*	*0.05*	*0.075*	*0.1*
C	*0.05*	*0.05*	*-*	*0.075*	*-*
D	*0.05*	*0.05*	*0.075*	*0.25*	*0.1*
E	*0.05*	*0.05*	*-*	*0.1*	*-*

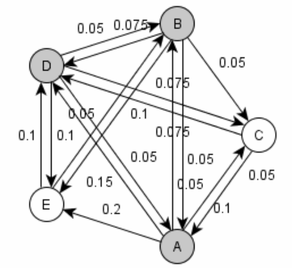


Fig. 8.26 *DocOG* for d_1 – OG_{d_1}

Table 8.18 Terms and relations in OG_{d_2}

	A	*B*	*C*	*D*	*E*
A	-	-	-	*0.225*	*0.1*
B	-	-	-	*0.225*	*0.1*
C	-	-	-	*0.225*	*0.1*
D	*0.075*	*0.15*	*0.225*	*0.75*	*0.3*
E	*0.025*	*0.05*	*0.075*	*0.3*	*0.25*

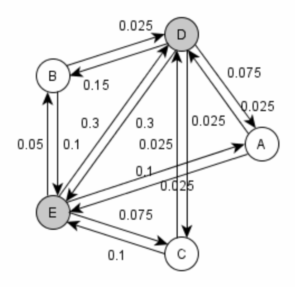


Fig. 8.27 *DocOG* for $d_2 - OG_{d_2}$

Step 4: Match the queries to the generated *DocOGs*

- q_1*-to-*OG_{d_1} (see Table 3.19), and q_1*-to-*OG_{d_2} (see Table 3.20)
- q_2*-to-*OG_{d_1} (see Table 3.21), and q_2*-to-*OG_{d_2} (see Table 3.22)
- q_3*-to-*OG_{d_1} (see Table 3.23), and q_3*-to-*OG_{d_2} (see Table 3.24)

Table 8.19 Result of q_1*-to-*OG_{d_1}

	A	*B*	*C*	*D*	*E*
A	-	*0.05*	-	-	-
B	*0.05*	*0.25*	*0.05*	*0.075*	*0.1*
C	-	*0.05*	-	-	-
D	-	*0.05*	-	-	-
E	-	*0.05*	-	-	-

Table 8.20 Result of q_1*-to-*OG_{d_2}

	A	*B*	*C*	*D*	*E*
A	-	-	-	-	-
B	-	-	-	*0.225*	*0.1*
C	-	-	-	-	-
D	-	*0.15*	-	-	-
E	-	*0.05*	-	-	-

Table 8.21 Result of q_2*-to-*OG_{d_1}

	A	*B*	*C*	*D*	*E*
A	-	-	*0.1*	-	-
B	-	-	*0.05*	-	-
C	*0.05*	*0.05*	-	*0.075*	-
D	-	-	*0.075*	-	-
E	-	-	-	-	-

Table 8.22 Result of q_2*-to-*OG_{d_2}

	A	*B*	*C*	*D*	*E*
A	-	-	-	-	-
B	-	-	-	-	-
C	-	-	-	*0.225*	*0.1*
D	-	-	*0.225*	-	-
E	-	-	*0.075*	-	-

Table 8.23 Result of q_3*-to-*OG_{d_1}

	A	*B*	*C*	*D*	*E*
A	-	-	*0.05*	-	*0.1*
B	-	-	*0.025*	-	*0.05*
C	*0.025*	*0.025*	-	*0.0375*	-
D	-	-	*0.0375*	-	*0.05*
E	*0.025*	*0.025*	-	*0.05*	-

Table 8.24 Result of q_3*-to-*OG_{d_2}

	A	*B*	*C*	*D*	*E*
A	-	-	-	-	*0.05*
B	-	-	-	-	*0.05*
C	-	-	-	*0.1125*	*0.05*
D	-	-	*0.1125*	-	*0.15*
E	*0.0125*	*0.025*	*0.0375*	*0.15*	*0.125*

Step 5: Calculate the score of each matching

- $score\ (q_1, OG_{d_1}) = 0.725$ and $score\ (q_1, OG_{d_2}) = 0.525$
- $score\ (q_2, OG_{d_1}) = 0.400,$ and $score\ (q_2, OG_{d_2}) = 0.625$
- $score\ (q_3, OG_{d_1}) = 0.500,$ and $score\ (q_3, OG_{d_2}) = 0.875$

Step 6: Calculate the similarity of each matching by Cosine similarity

- $score\ (q_1, OG_{d_1}) = 0.336,$ and $score\ (q_1, OG_{d_2}) = 0.169$ / Result: $d_1 > d_2$
- $score\ (q_2, OG_{d_1}) = 0.185,$ and $score\ (q_2, OG_{d_2}) = 0.201$ / Result: $d_2 > d_1$
- $score\ (q_3, OG_{d_1}) = 0.348,$ and $score\ (q_3, OG_{d_2}) = 0.422$ / Result: $d_2 > d_1$

Summary of the querying result

For the querying result of q_1, document d_1 is more relevant than document d_2, this is normal because the term *B* appears in document d_1 but not in d_2. For the querying result of q_2, document d_2 is more relevant than document d_1 although both documents do not contain the term of the query (term *C*). This is because term *C* is more related to the terms *D* and *E* than the terms *A* and *B* in the domain (as measured in the domain ontology graph OG_A), and document d_1 contains mainly terms *A* and *B* while document d_2 contains mainly terms *D* and *E*, therefore document d_2 is more relevant than document d_2. For the querying result of q_3, document d_2 is more relevant than document d_1 although both documents contain exactly one term only in the query (d_1 contains term *B* and d_2 contains term *E*). Document d_2 is more relevant because term *E* is weighted higher than term *B* in the domain, i.e. term *E* is more important in the domain and also has higher relations to other terms, comparing to the term *B*. This querying example shows that the Ontology Graph based document retrieval method is not only relying on exact term matching, but also taking consideration of the term-relationship to other terms. So that the retrieval result of a query does not return only documents which contain the terms in that query, but also returns documents which do not contain the terms and may be relevant to that query. This can enhance the performance of some traditional search engines that use the keyword-based matching retrieval method, by inputting a domain knowledge that can describe the related concepts about the domain.

Part III

KnowledgeSeeker: Applications

Chapter 9
Ontology Graph Based Approach for Automatic Chinese Text Classification

Abstract. Automatic classification of Chinese text documents requires a machine to process and analyze the meaning of Chinese terms. We propose an Ontology Graph based approach to measure the relations between Chinese terms for the text classification purpose. The method improves traditional high dimensional term-based text classification approach, in that the new method selects very small number of semantically related concepts to create Ontology Graphs. The Ontology Graphs can be used to represent different classes (domains). It enhances text classification performance by using its small-size but high semantically associated concepts. Our experiments show that the proposed method has classified a Chinese document set with 92% accuracy in f-measure by using Ontology Graphs containing only 80 concepts for each class. The high accuracy result shows that the Ontology Graphs used in the process are enable to represent the knowledge of a domain and also the Ontology Graph based approach of text classification is effective and accurate.

9.1 Introduction

Automatic text classification is a process in which a machine analyzes the content of a document. A variety of machine-learning approaches are currently used for this task including *tf-idf*, support vector machines (SVM), and the *k-NN* approach, all of which work by measuring the frequency of words in a document. An obvious drawback of such approaches is that, from the viewpoint of text classification, a measurement of frequency is by no means a measurement of importance and frequency-based approaches therefore give too much weight to intuitively less important words, especially in English with its abundance of function and grammatical words (articles, auxiliary verbs, etc). In this sense, low-relevance words are a type of noise and as such affect both classification speed and accuracy while contributing little of value (Zheng et al. 2003). One way to remove high-frequency, low-classification-value words is to apply feature selection (or feature reduction), traditionally either supervised or unsupervised. Supervised feature selection

E.H.Y. Lim, J.N.K. Liu, and R.S.T. Lee: Knowledge Seeker, ISRL 8, pp. 145 – 164.
springerlink.com

methods based on information gain (IG) and Chi-square (χ^2) have been shown [Li et al. 2008] to do well in text classification and clustering. However, such approaches are not as effective when applied to Chinese text classification. This is in part because Chinese features many fewer function and grammatical words than English and so reduction can remove words that may be important in classification. Further, word disambiguation is not as straightforward in Chinese, which does not use a space between individual words as in English, and so there are considerable difficulties associated with defining, identifying, and extracting word accurately when using frequency-based approaches.

There are a number of text classification approaches that are less dependent on classification by frequency, including the vector space model (Aggarwal et al. 2004), ontology based model (Lim et al. 2008) and Chi-square statistic (Li et al. 2008). The vector space model uses a term vector to represent every document and topic class, giving a score (or value) to each term in the vector, calculating the weight of terms inside a document using a scoring function such as the commonly used *tf-idf* (Aggarwal et al. 2004) and classifying texts by comparing the document vector and class vector. Research (Li et al. 2008) has shown that this method is about as accurate as approaches such as neural networks and *k-NN*. Ontology based text mining (Rezgui 2007) operates by using a machine understandable knowledge to analyze text documents. The knowledge (ontology) is either created manually or semi-automated by machine learning approaches. Previous research (Lim et al. 2008) has combined an agent -based ontological system and a vector space model to retrieve and analyze Chinese texts. The ontology was based on an existing Chinese dictionary, HowNet (Dong 2003), and relations between terms were calculated based on the structure defined in the HowNet. While highly accurate, a drawback of this approach is that the ontologies upon which the learning algorithms depend are manually-constructed and automatic or even semi-automatic ontology construction remains a difficult task. χ^2 based feature selection is a statistical measure that is used to calculate term-class interdependence (Mesleh 2007), analyzing the relationship between a term and a class (a topic or a category). Previous research has shown that χ^2 statistic based supervised feature selection method can improve text classification and clustering performance when a class-labeled corpus is available. Two variants of the χ^2 statistic are correlation coefficient and GSS coefficient (Busagala et al. 2008) while (Li et al. 2008) has proposed a new supervised feature selection method that is an extension of χ^2 and is used to measure an either positive or negative relationship between a term and a class.

In this chapter we propose a novel approach that uses Ontology Graph for text classification, in that the Ontology Graph is generated based on the χ^2 statistic. Unlike traditional χ^2 measurement, however, which measures the degree to which a term is related to a particular domain (class), the Ontology Graph based approach also measures the degree to which terms are dependent on other terms. The domain Ontology Graph learning and generation methods have been discussed in Chapters 6 and 7. We further apply the Ontology Graph querying methods which have been discussed in Chapter 8, together with an algorithm based on vector space model, to measure how a Chinese text document is related to each Ontology Graph for classification purpose. Our experimental results show that the Ontology Graph based approach is highly effective when processed in text classification (92% accuracy in f-measure by using Ontology Graphs containing only 80 concepts for each class).

9.2 Methodologies

We describe the methodologies by first reviewing the theory of Ontology Graph model, and then we describe the classification algorithm which integrates the vector space model, Ontology Graph model, and Ontology Graph based comparison method.

9.2.1 Ontology Graphs Reviews

We define Ontology Graph as a set of concepts, in which concepts are created by a set of terms and relations between them. The relations of terms are enhances by weight, which is generated automatically by a χ^2 statistic based method, for representing how close of two terms are related. Figure 9.1 visualizes the conceptual structure of an Ontology Graph:

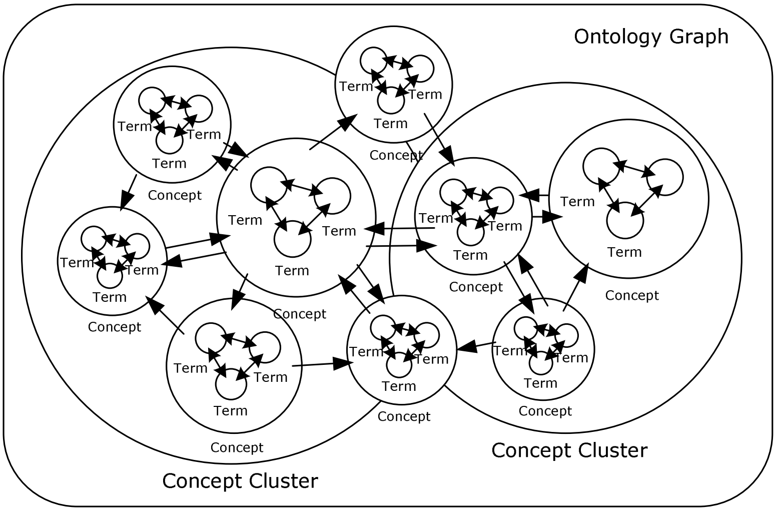


Fig. 9.1 Conceptual structure of Ontology Graph

The formal definition of Ontology Graph is defined as:

$$OG_d = <T, F, H, R, C, A>$$

- d defines the domain of the Ontology Graph is associated with
- T is a set of terms t_i of OG_d
- F is a set of word functions of terms $t_i \in T$
- H is a set of taxonomy relationships of T
- R is a set of relations between t_i and $t_{j,}$ where t_i , $t_j \in T$
- C is a set of clusters of $t_{i,}\ldots,t_{n,}$ where $t_1,\ldots,t_n \in T$
- A is a set of axioms that characterize each relation of R

9.2.2 Classification Algorithm

The text classification algorithm represents every document by a term-frequency vector $TF = < tf_1, tf_2, ..., tf_n >$, for n-dimension term-space according to the number of terms created in all *DOGs*. Each document which is represented by a term-frequency vector is then compared to every domain ontology graph as to measure their similarity. The document is assigned to a domain class if the comparison of that document to the corresponding *DOG* gets the highest similarity value (score). Therefore, one document may belong to multiple classes with different weights according to its score, but in this classification algorithm we choose to assign one document to a single class according to the highest similarity value measured.

The comparison is done by matching every text document to *DOG* (the process described in Chapter 8). Therefore, if there is m number of *DOGs* (m domains to be classified), every document is be compared m times to find out the highest similarity values (Figure 9.2). The detailed matching and calculation process has been discussed in Chapter 8, the major comparison methods are presented as follows:

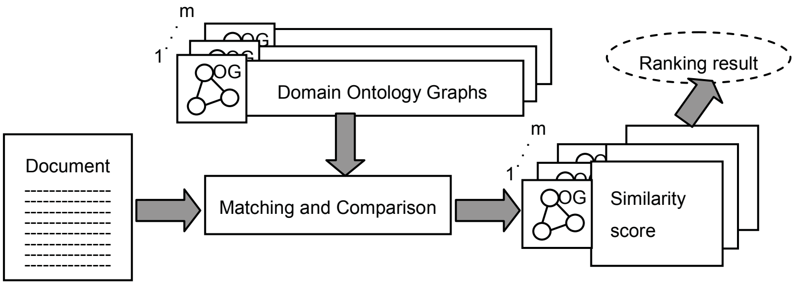


Fig. 9.2 Comparison of document and domain ontology graph for similarity measurement

Comparison Methods

The comparison relies on a score function that scores the terms in a document, for those terms also appear in the compared Domain Ontology Graph (Figure 9.3).

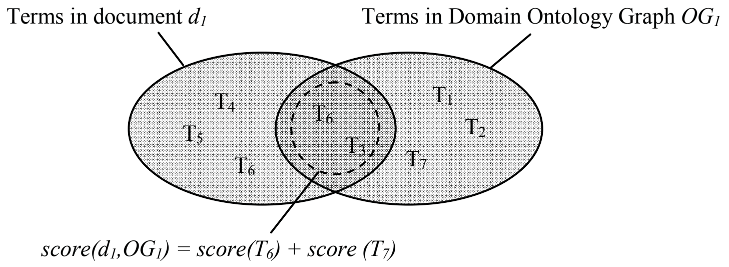

Terms in document d_1
Terms in Domain Ontology Graph OG_1
T_4
T_5
T_6
T_6
T_3
T_1
T_2
T_7
$score(d_1, OG_1) = score(T_6) + score\ (T_7)$

Fig. 9.3 Scoring terms inside a document

Example 9.1 – Ontology Graph Based Text Classification

The text classification process combines the Ontology Graph matching and comparison process described in Chapter 8. The classification relies on a score function that scores a document to every *DOGs*, so that we can select the highest scored *DOG* matching as the classified domain (class).

Step 1: Generates *DocOGs* by matching the documents to every *DOG* (class)

- If there are *m* classes of domain to be classified, *m* number of *DOGs* are created:

  $OG_1, OG_2, \ldots, OG_m$ *for classes* $C = \{c_1, c_2, \ldots, c_m\}$

- Generate *m* number of *DocOGs* correspondingly to each *DOG*:

  d_1 → *[Generation Process (refer to Chapter 8)]* → $OG_{d_1,1}, OG_{d_1,2}, \ldots, OG_{d_1,m}$

Step 2: Obtain the scores of vectors of every *DocOG*

- $scores(d_1) = \{<c_1, s_1>, <c_2, s_2>, \ldots, <c_m, s_m>\}$ *where*

  $s_i = score(OG_{d_1}, OG_i)$ *for* $i \in \{1 \ldots m\}$. *(refer to Chapter 8)*

Step 3: Select the highest scored *DocOG* as the classified domain

- *Classified class =* c_j *for* $MAX(s_j)$

9.3 Experiments

The text classification experiment to be described here has two purposes. On one hand we wish to evaluate the classification performance of the proposed Ontology Graph based approach by comparing to other classification methods. On the other hand, we wish to determine the optimal size (number of terms) of *DOG* for a class which can produce the best classification result.

9.3.1 Experiments Description

9.3.1.1 Evaluate the Performance of Ontology-Graph Based Approach (Experiment 1)

The first experiment presents a text classification case by using three different approaches to classify documents. The first one is the traditional *tf-idf* approach. The second one is the term-dependency approach which replaces the *tf-idf* weight by the term-dependency (*R*) weight in *DOG*. The third one is the ontology-graph approach which scores a document to a class by the weight of relationships between each concept in the Ontology Graph. We aim to evaluate and compare the performance of different text classification approaches in terms of its accuracy (recall/precision). The three different text classification approaches are described as follows:

1. Term frequency-inverse document frequency (tf-idf) approach

This approach uses a scoring function that scores the terms occurred in the document by the term frequency and the inverse document frequency. This scoring function is the same as the traditional *tf-idf* classification approach and it is defined as:

$$score(t_i) = tf_{t_i} \times idf_{t_i}$$

2. Term-dependency (R) approach

This approach uses a scoring function that scores the terms occurred in the document by the term weight in the Domain Ontology Graph (*DOG*). Term weights in the Ontology Graph are represented by the dependency measurement – *R*, and it is calculated in the Ontology Graph learning process. The term-dependency scoring function is defined as:

$$score(t_i) = tf_{t_i} \times R_{t_i}$$

3. Ontology-graph approach

The ontology-graph based text classification approach is processed by matching a Document Ontology Graph (*DocOG*) to Domain Ontology Graph (*DOG*). The algorithm has been presented in Chapter 9.2. The scoring function for comparing a document to a *DOG* is defined as:

$$score(d_1, OG_1) = score(OG_{d_1}, OG_1)$$

9.3.1.2 Evaluate the Optimum Size of Domain Ontology Graph for the Best Classification Result (Experiment 2)

The second experiment presents an extended text classification case by using those three different approaches presented in experiment 1 and further varying the size of dimensions of terms used in each approach. The process used the same scoring functions presented in experiment 1 and tried to apply them into different sizes of class vector (for *if-idf* and *term-dependency* approaches) or sizes of Domain Ontology Graph (for *ontology-graph* approach) to do the text classification. In this experiment, we aimed to evaluate how the size of terms in each approach affects the classification performance

9.3.1.3 Evaluate the Effects for Setting Different Thresholds of Weight of Domain Ontology Graph (Experiment 3)

The third experiment presents a text classification case that uses only the ontology-graph approach. It is carried out by using a fixed size of domain ontology graphs but varying the threshold of weight between concepts in those domain ontology graphs. A higher threshold value reduces the number of relationships between concepts in each ontology graph. The size of each domain ontology graph is therefore further reduced in this case. In this experiment, we aimed to evaluate how the thresholds of weight (i.e. the number of concepts' relationship) affect the classification performance.

9.3.2 Evaluation Method

Error rate is the most practical measurement to evaluate the information retrieval model. This measurement is aimed to calculate the retrieval accuracy, in terms of precision, recall, and f-measure. It is done by first observing the retrieval correctness from the result, as shown in Table 9.1:

Table 9.1 The table of retrieval result

	Relevant	Non-relevant
Retrieved	*TP*	*FN*
Not retrieved	*FP*	*TN*

- *TP* (True Positive) – the number of relevant documents, retrieved as relevant
- *FP* (False Positive) – the number of relevant documents, not retrieved as relevant
- *FN* (False Negative) – the number of non relevant documents, retrieved as non relevant
- *TN* (True Negative) – the number of non relevant documents, not retrieved as relevant.

Performance measurement

Precision – It measures the accuracy of the retrieval model, by calculating the percentage of correctly retrieved documents to the whole retrieved result set. It is defined by:

$$precision = \frac{TP}{TP + FP}$$

Recall – It measure the ability of the retrieval model to retrieve correct documents from the whole data set, by calculating the percentage of correctly retrieved document to all the documents that should be retrieved. It is defined by:

$$recall = \frac{TP}{TP + FN}$$

F-measure – It measures the harmonic average of precision and recall. It is defined by:

$$f - measure = \frac{2 \times precision \times recall}{precision + recall}$$

9.3.3 Performance on Ontology Graph Based Text Classification

Experimental Data Sets

Data set required for the experiment mainly includes 1. A training document sets for learning and generating domain ontology graphs, and 2. A testing document set for automatic text classification and performance evaluation.

Training and Testing Document Sets

The training document sets are used for learning and generating domain ontology graphs. The training document is a labeled document corpus, i.e. all documents are classified into a specific label of class. Each class label represents a particular domain which is equivalent to the corresponding generated domain ontology graph. The training documents are classified into 10 classes and they are shown in Table 9.2.

Table 9.2 Class label (Chinese & English) of training document set

Class	Class Label (Chinese)	Class Label (English)
1	文藝	Arts and Entertainments
2	政治	Politics
3	交通	Traffic
4	教育	Education
5	環境	Environment
6	經濟	Economics
7	軍事	Military
8	醫療	Health and Medical
9	電腦	Computer and Information Technology
10	體育	Sports

Number of documents in the training set and testing set are shown in Table 9.3. Training set is used for domain ontology graph learning and testing set is used for evaluation purpose. Training set contains 1972 documents (70% of the whole) and testing set contains 842 documents (30% of the whole).

Table 9.3 Document count for training and testing document sets

Class	Class Label	Training Set	Testing Set
1	文藝	174	74
2	政治	354	151
3	交通	150	64
4	教育	154	66
5	環境	141	60
6	經濟	228	97
7	軍事	174	75
8	醫療	143	61
9	電腦	139	59
10	體育	315	135
	Total	1972 (70% of 2814)	842 (30% of 2814)

9.3.4 Experimental Results

This section provides the results of different experiments described in section 9.3.3. We describe the evaluation methods first and then present the detailed experimental results.

9.3.4.1 Performance on Ontology Graph Based Text Classification (Experiment 1)

Precision and recall values have been computed for the three classification approaches. Table 9.4 shows the detailed result obtained by computing precision, recall and f-measure for the three approaches by using a term-size of 30, i.e. 30 of terms are used in the *tf-idf* method and 30-term sized *DOG* (as presented in Chapter 6.1.5) are used in the *term-dependency* and *ontology-graph* method. The table shows the precision, recall, and f-measure of each class in the test document set, and also its average. Figure 9.4 depicts the comparison of different scoring methods in precision, Figure 9.5 depicts the comparison of different scoring methods in recall, and Figure 9.6 depicts the comparison of different scoring methods in f-measure.

Table 9.4 Details of classification result of each class

Class	Approach	Precision	Recall	F-measure
文藝 (Arts and Entertainments)	*tf-idf*	0.9426	0.8243	0.8795
	term-dependency	0.9306	0.9054	0.9178
	ontology -graph	0.9333	0.9200	0.9266
政治 (Politics)	*tf-idf*	0.7165	0.9146	0.8035
	term -dependency	0.6032	0.9868	0.7487
	ontology -graph	0.8544	0.8940	0.8738
交通 (Traffic)	*tf-idf*	0.9831	0.9063	0.9431
	term -dependency	0.9825	0.8750	0.9256
	ontology -graph	0.9355	0.9063	0.9206
教育 (Education)	*tf-idf*	0.9649	0.8333	0.8943
	term -dependency	0.9836	0.9091	0.9449
	ontology -graph	0.9118	0.9394	0.9254
環境 (Environment)	*tf-idf*	0.8727	0.8000	0.8348
	term -dependency	0.9245	0.8167	0.8673
	ontology -graph	0.9483	0.8800	0.9129
經濟 (Economics)	*tf-idf*	0.6071	0.8763	0.7173
	term -dependency	0.8191	0.7938	0.8063
	ontology -graph	0.8058	0.8557	0.8300
軍事 (Military)	*tf-idf*	0.9111	0.5467	0.6833
	term -dependency	0.9756	0.5333	0.6897
	ontology -graph	0.8571	0.7546	0.8026
醫療 (Health and Medical)	*tf-idf*	0.9556	0.7049	0.8113
	term -dependency	1.0000	0.7049	0.8269
	ontology -graph	0.9792	0.7705	0.8624
電腦 (Computer and Information Technology)	*tf-idf*	0.9600	0.8136	0.8807
	term -dependency	0.9808	0.8644	0.9189
	ontology -graph	0.8730	0.9257	0.8986
體育 (Sports)	*tf-idf*	0.9474	0.9106	0.9286
	term -dependency	0.9918	0.8963	0.9416
	ontology -graph	0.9771	0.9256	0.9507
Average	*tf-idf*	0.8861	0.8130	0.8480
	term -dependency	0.9192	0.8286	0.8715
	ontology -graph	0.9076	0.8772	0.8921

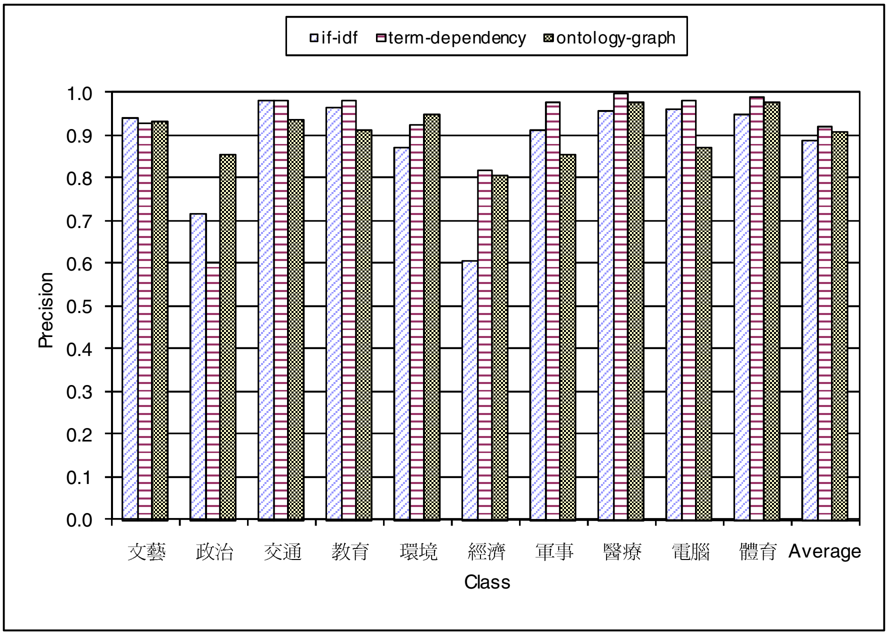


Fig. 9.4 Result of precision for different approaches

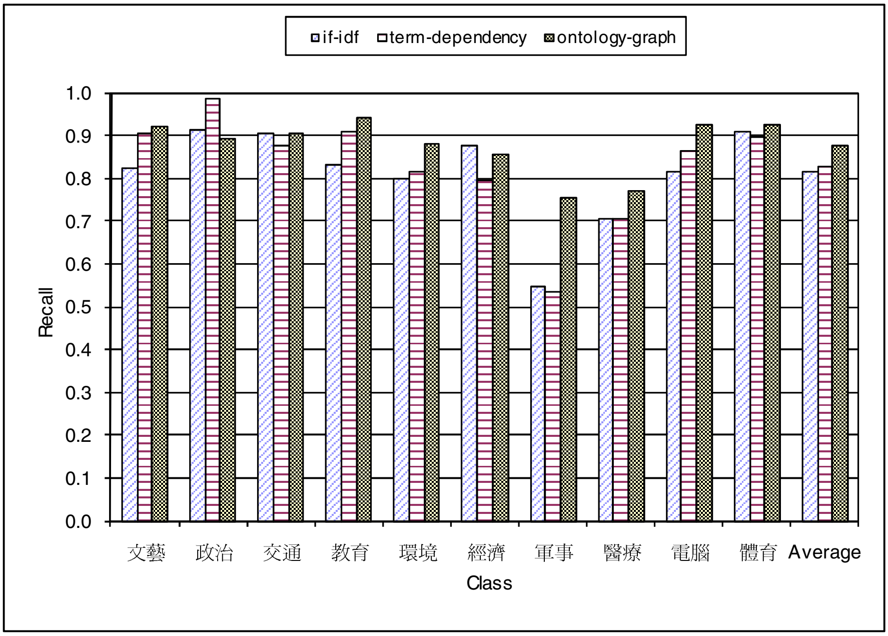


Fig. 9.5 Result of recall for different approaches

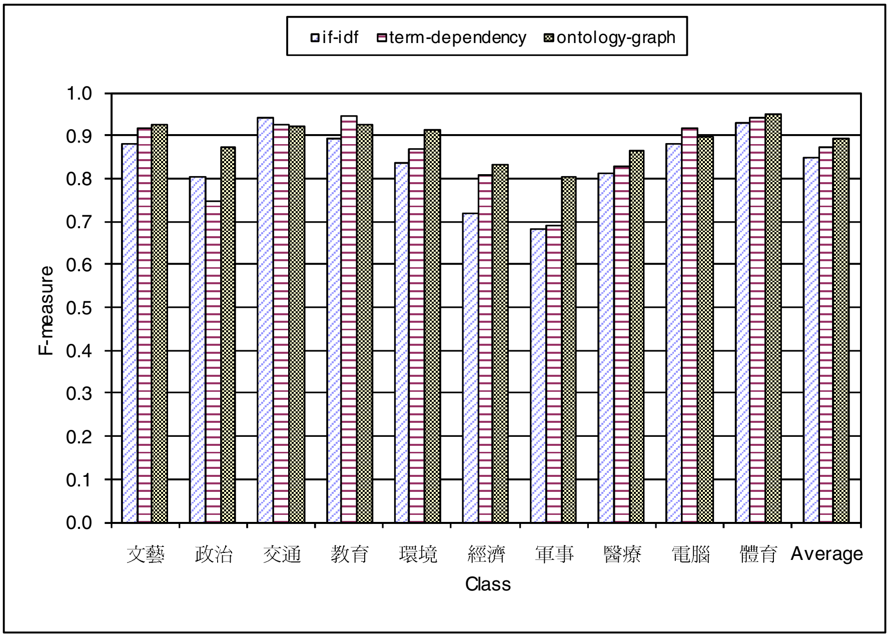


Fig. 9.6 Result of F-measure for different approaches

The above experimental result has shown that the *ontology-graph* approach performs the highest classification accuracy (89.2% of f-measure). The *term-dependency* method performs the second highest classification accuracy (87.2% of f-measure), while the *tf-idf* performs the lowest classification accuracy (84.8% of f-measure) among the three methods have been tested. This experiment has shown that the *DOGs* are useful to represent a domain of classes and also it is useful to develop a classification system. By comparing to the *term-dependency* method, it revealed that the relationship of concepts in the ontology graph is useful to represent knowledge. This is because using the relationship information in *DOG* (*ontology-graph* approach) to do the text classification performs better result than not using the relationship (*term-dependency* approach). Therefore, this concludes that the *ontology-graph* approach is an effective approach for developing a text classification system.

9.3.4.2 Performance on Using Different Size of Terms (Dimensionality) for Text Classification (Experiment 2)

In the previous experiment, we have found that the ontology-graph approach performs the best in text classification among all three tested approaches. In this experiment, we further evaluate those three methods by varying the size of terms (the number of term nodes in *DOG*) used in the text classification process. The precision and recall values have been computed for this experiment by using different sizes of term nodes of *DOGs*. The sizes of the term nodes in *DOGs* tested in this experiment are: 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 150, 200, and 300. Tables 9.5, 9.6 and 9.7 give the classification result of the experiments for the three approaches – *tf-idf,* term-dependency, and ontology-graph correspondingly, presenting the precision, recall, and f-measure values of the result. Figures 9.7 to 9.12 depict their result in graphical format.

Result of using *tf-idf* approach

Table 9.5 Classification result for *tf-idf* approach

Size	Precision	Recall	F-measure
10	0.8396	0.8011	0.8199
20	0.8723	0.8119	0.8410
30	0.8861	0.8130	0.8480
40	0.8838	0.8162	0.8487
50	0.8877	0.8261	0.8558
60	0.9002	0.8372	0.8676
70	0.8957	0.8286	0.8608
80	0.9050	0.8214	0.8612
90	0.9010	0.8237	0.8606
100	0.8986	0.8157	0.8551
150	0.9031	0.804	0.8506
200	0.8982	0.7962	0.8441
300	0.9034	0.7912	0.8436

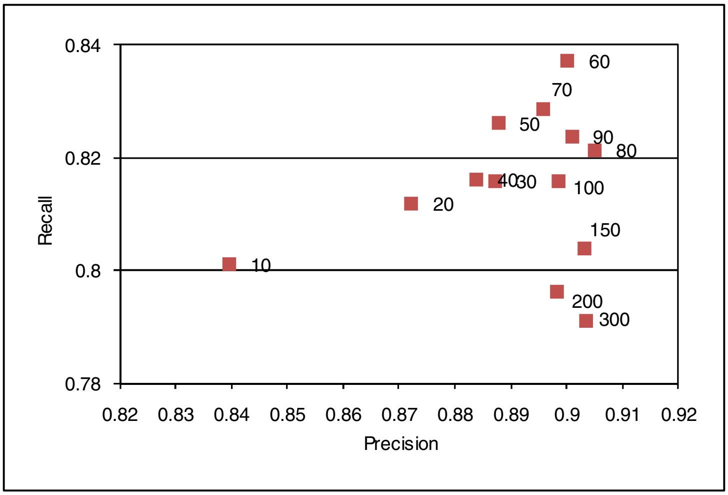


Fig. 9.7 Result of precision and recall for *tf-idf* approach

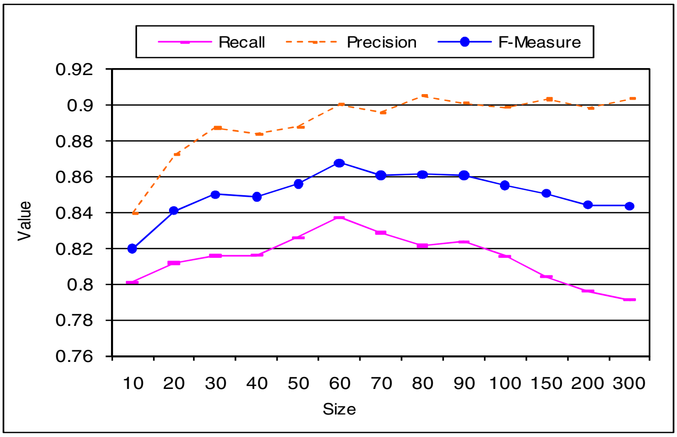


Fig. 9.8 Result of precision, recall and f-measure for *tf-idf* approach

Result Description

As shown in Table 9.5, the *tf-idf* approach for the text classification gives accuracy in f-measure in ranges 82% and 86.8%. Using the term-size of 10 gives the lowest precision (84.0%) and using the term-size of 80 gives the highest precision (90.5%). Using the term-size of 300 gives the lowest recall (79.1%) and using the term-size of 60 gives the highest recall (83.7%). Using the term-size of 10 gives the lowest f-measure (82.0%) and using the term-size of 60 gives the highest f-measure (86.8%).

Result of using term-dependency approach

Table 9.6 Classification result for term-dependency approach

Size	Precision	Recall	F-measure
10	0.9061	0.7657	0.8300
20	0.9130	0.8016	0.8537
30	0.9192	0.8286	0.8715
40	0.9123	0.8310	0.8697
50	0.9107	0.8400	0.8739
60	0.9087	0.8370	0.8714
70	0.9138	0.8389	0.8747
80	0.9187	0.8466	0.8812
90	0.9136	0.8460	0.8785
100	0.9196	0.8544	0.8858
150	0.9162	0.8544	0.8842
200	0.9177	0.8548	0.8851
300	0.9206	0.8597	0.8891

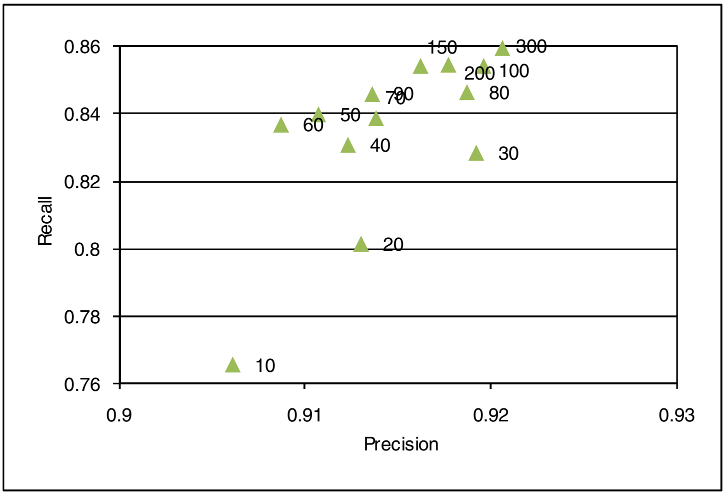


Fig. 9.9 Result of precision and recall for term-dependency approach

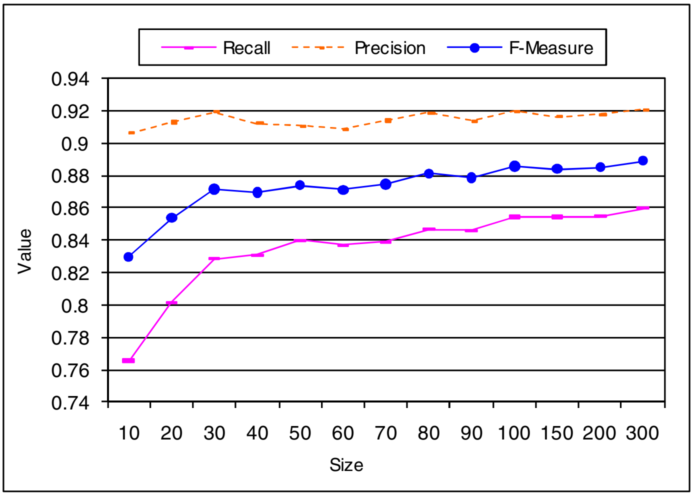


Fig. 9.10 Result of precision, recall and f-measure for term-dependency approach

Result Description

As shown in Table 9.6, the *term-dependency* approach for the text classification gives accuracy in f-measure in ranges 83% and 88.9%. Using the term-size of 10 gives the lowest precision (90.6%) and using the term-size of 300 gives the highest precision (92.1%). Using the term-size of 10 gives the lowest recall (76.6%)

and using the term-size of 300 gives the highest recall (86.0%). Using the term-size of 10 gives the lowest f-measure (83.0%) and using the term-size of 300 gives the highest f-measure (88.9%).

Result of using *ontology-graph* scoring approach

Table 9.7 Classification result for ontology-graph approach

Size	Precision	Recall	F-measure
10	0.9023	0.8329	0.8662
20	0.9116	0.8631	0.8867
30	0.9076	0.8772	0.8921
40	0.9102	0.8846	0.8972
50	0.9213	0.8913	0.9061
60	0.9239	0.8914	0.9074
70	0.9325	0.9078	0.9200
80	0.9360	0.9103	0.9230
90	0.9325	0.9078	0.9200
100	0.9293	0.9054	0.9172
150	0.9240	0.9039	0.9138
200	0.9226	0.9015	0.9119
300	0.9254	0.9035	0.9143

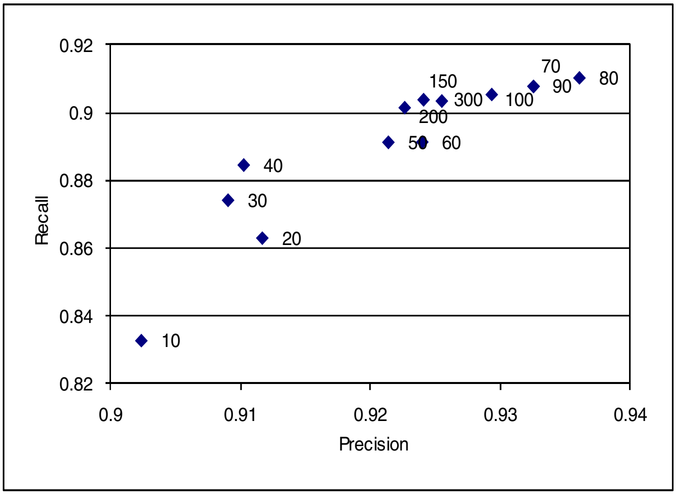


Fig. 9.11 Result of precision and recall for ontology-graph approach

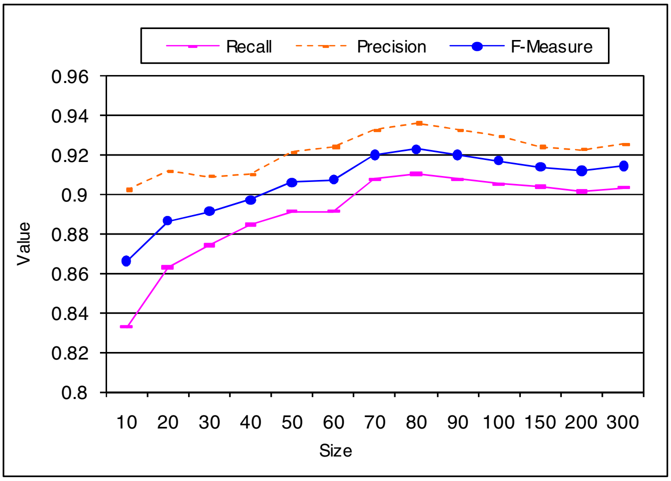


Fig. 9.12 Result of precision, recall and f-measure for ontology-graph approach

Result Description

As shown in Table 9.7, the *ontology-graph* approach for the text classification gives accuracy in f-measure in ranges 86.6% and 92.3%. Using the size of ontology graph of 10 gives the lowest precision (90.2%) and using the size of ontology graph of 80 gives the highest precision (93.6%). Using the size of ontology graph of 10 gives the lowest recall (83.3%) and using the size of ontology graph of 80 gives the highest recall (91.0%). Using the size of ontology graph of 10 gives the lowest f-measure (86.6%) and using the size of ontology graph of 80 gives the highest f-measure (92.3%).

9.3.4.3 Result of Using Different Thresholds of Weight of Domain Ontology Graphs (Experiment 3)

In the previous experiment, we have evaluated that using the size of 80 of the domain ontology graphs can obtain the optimized performance in the text classification accuracy. In this experiment, we fixed the size of ontology graphs (nodes) to 80, and further evaluate how the sizes of edges of domain ontology graphs affect the performance of the text classification process. We use the threshold θ_v to filter dependency edges of domain ontology graphs. The threshold value has been presented in Chapter 7.3. Edges with weight lower than the threshold are removed, and these edges are then excluded in the calculation in ontology-graph classification approach. Therefore, higher threshold values decrease the number of edges of an ontology graph, and thus further reduced the size of the ontology graph. The threshold values evaluated in this experiment are 0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, and 1.0. Table 9.8 shows the result of the experiment, which shows the precision, recall, and f-measure values of the result. Figures 9.13 and 9.14 depict the result in graph for comparison.

Table 9.8 Experimental result of using different threshold values

Threshold	Precision	Recall	F-measure
0.0	0.9361	0.9104	0.9230
0.1	0.9249	0.8940	0.9092
0.2	0.9141	0.8830	0.8983
0.3	0.9041	0.8728	0.8882
0.4	0.8793	0.8370	0.8576
0.5	0.8597	0.8064	0.8322
0.6	0.8329	0.7590	0.7942
0.7	0.8058	0.7039	0.7514
0.8	0.7895	0.6557	0.7164
0.9	0.7784	0.6038	0.6801
1.0	0.7784	0.5738	0.6606

As shown in Table 9.8 and Figures 9.13 – 9.14, higher threshold values of weight used for the text classification result in lower precision and recall. If the threshold value is set to 1.0, the f-measure drops to about 0.66. If the threshold value is set to 0.0 (i.e. no threshold, and every edge is used for ontology-graph scoring), the f-measure retains at about 0.923. Therefore, the threshold affects the classification performance if the threshold value is set. No threshold value can obtain the best performance. The best classification performance is 0.93 in precision, 0.91 in recall, and 0.92 in f-measure with the threshold equal to 0.

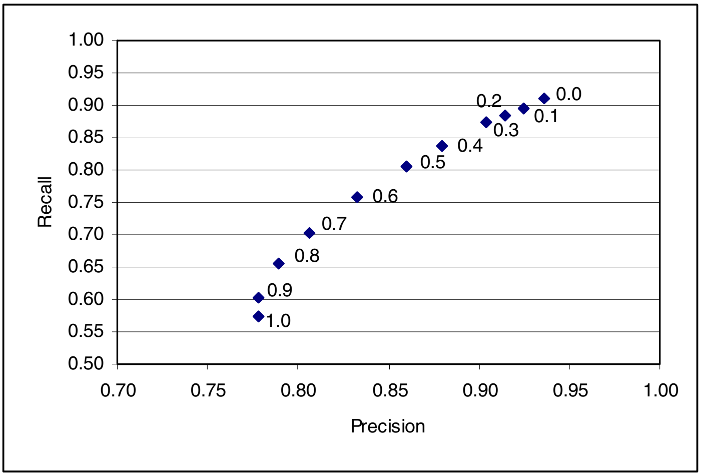


Fig. 9.13 Precision and recall for different threshold values used

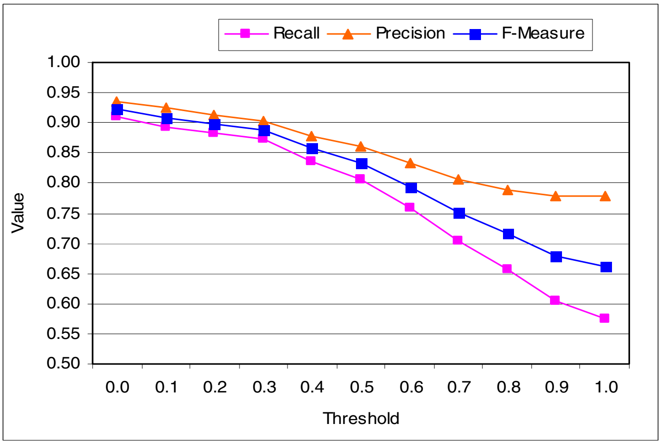


Fig. 9.14 Result of classification in precision, recall and f-measure

9.3.4.4 Combining the Results and Optimizing the Parameters for the Text Classification Experiments

This result is to combine the previous experiments to figure out an optimal setting for the text classification process. We found that the Ontology Graph is the best approach for implementing the text classification. In addition, a size of 80 terms of *DOG* gives the best performance. In the following figures, we show the combined result of the previous experiments for comparison purpose. We can see that the comparison result of precision and recall (Figure 9.15), and the comparison result of f-measure by using different sizes of terms (Figure 9.16).

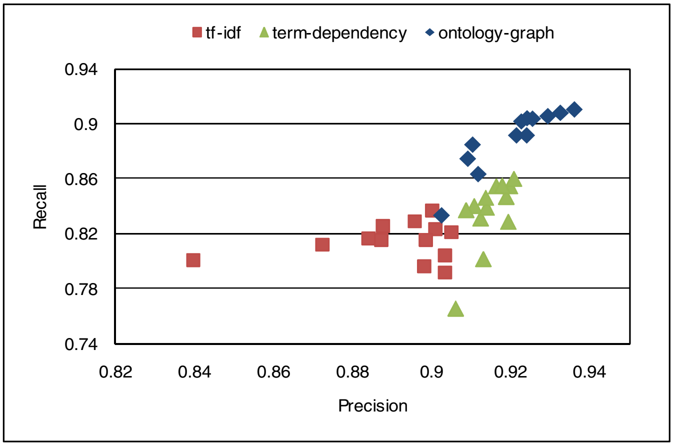


Fig. 9.15 Result comparison of precision and recall for the three approaches

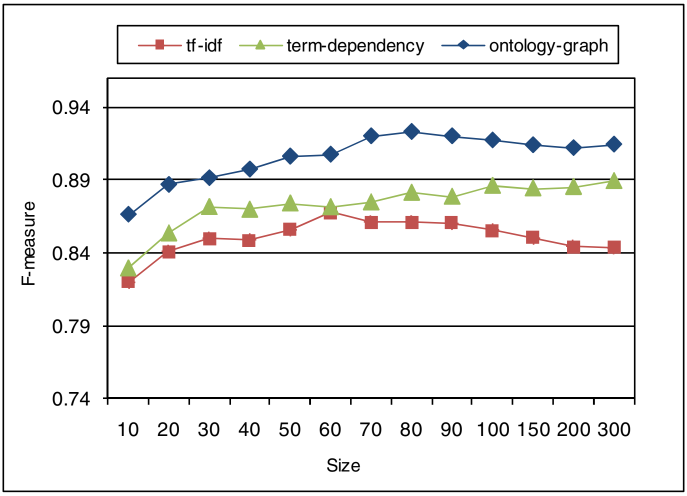


Fig. 9.16 Result comparison of f-measure for the three approaches

Result Conclusion

As shown in Figures 9.15 to 9.16, the use of *Ontology Graph* approach performs the best for every term-size used. Generally, for the Ontology Graph based text classification approach, the use of smaller sizes of Ontology Graph results in lower precision and recall. However, the result also shows that the precision and recall are optimized by using the size of 80, size larger than 80 cannot increase the accuracy. Figure 4.22 also shows that the performance of the text classification system is optimized by using the Ontology Graph approach and by using 80 as the term-size of *DOGs*. Table 9.9 summarizes the details of the experimental results.

Table 9.9 Summary of the optimized performance

	Size for optimized Precision	Size for optimized recall	Size for optimized f-measure
tf-idf	80 (90.5%)	60 (83.7%)	60 (86.8%)
Term-dependency	300 (92.1%)	300 (86.0%)	300 (88.9%)
Ontology-graph	80 (93.6%)	80 (91.0%)	80 (92.3%)

Chapter 10
IATOPIA iCMS KnowledgeSeeker – An Integrated Content Management System and Digital Asset Management System (DAMS)

Abstract. IATOPIA iCMS KnowledgeSeeker is an integrated solution which has adopted the KnowledgeSeeker technology to develop various ontology based application, such as the IATOPIA Digital Asset management System. IATOPIA DAMS provides a centralized databank to categorize, manage, store and retrieve different types of digital asset, i.e. text articles, photos, videos and audio data. With IATOPIA patented Ontology System, users can define their own concept tree(s) to annotate (tagging) the attributes for all digital assets which can be used for different web channels, e-archive systems and search with IATOPIA patented ontology-based search engine.

10.1 IATOPIA iCMS KnowledgeSeeker

IATOPIA integrated Content Management System (iCMS) is an integrated and patented solution designed and implemented for IATOPIA.com limited. It provides solution for different content providers such as publishers, media, new agencies, libraries to organize, manage, search, data-mining, archive and retrieve their digital assets (e.g. news articles, photos/images, videos, audio clips) from IATOPIA patented centralized iCMS databank. With the integration of IATOPIA iCMS and the ontological KnowledgeSeeker system, all digital contents can be enhanced and organized by ontology based knowledge. The digital contents can be retrieved and disseminated through different channels such as IATOPIA Web Channels, IATOPIA e-publications, and mobile applications including iPhone and Windows Mobile.

10.1.1 System Features

IATOPIA iCMS KnowledgeSeeker is an ontological system that is used to manage and organize all digital content inside the iCMS by using ontology approach.

E.H.Y. Lim, J.N.K. Liu, and R.S.T. Lee: Knowledge Seeker, ISRL 8, pp. 165 – 173.
springerlink.com

The IAOPITA iCMS KnowledgeSeeker consists of a content databank cluster, an ontology index databank, and an IATOPIA ontological search engine. iCMS KnowledgeSeeker search engines use ontology approach to analyze Chinese text content (such as news articles), and also use the concept of semantic web to organize information semantically. iCMS KnowledgeSeeker also uses the ontology approach to identify the article topics (a text classification process). It has been tested and experimented with high performance, and has shown that it is a practical approach for using ontology technology to develop the search engine model.

10.1.2 System Model and Architecture

The IATOIPA iCMS KnowledgeSeeker consists of three components (Figure 10.1), the process flow between those components is shown in Figure 10.2:

1. IATOPIA Ontology and Content Index – it stores all ontology information and all analyzed information about all iCMS contents, including the ontology based index.
2. IATOPIA Ontological Search Engine – it integrates the process of content analysis, content indexing, index searching, and responses to user with the search result.
3. IATOPIA iCMS Databank Cluster – it stores all the original sources of content files, including article, audio, video, images, e-publication data, etc.

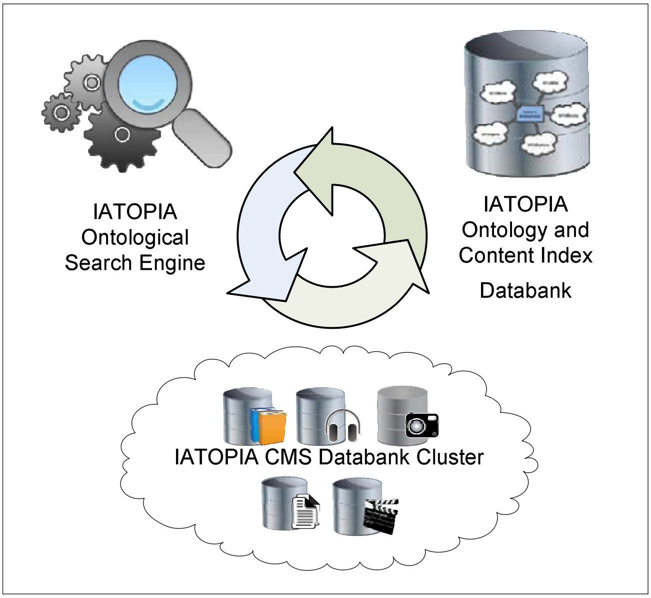


Fig. 10.1 The system architecture of iCMS KnowledgeSeeker

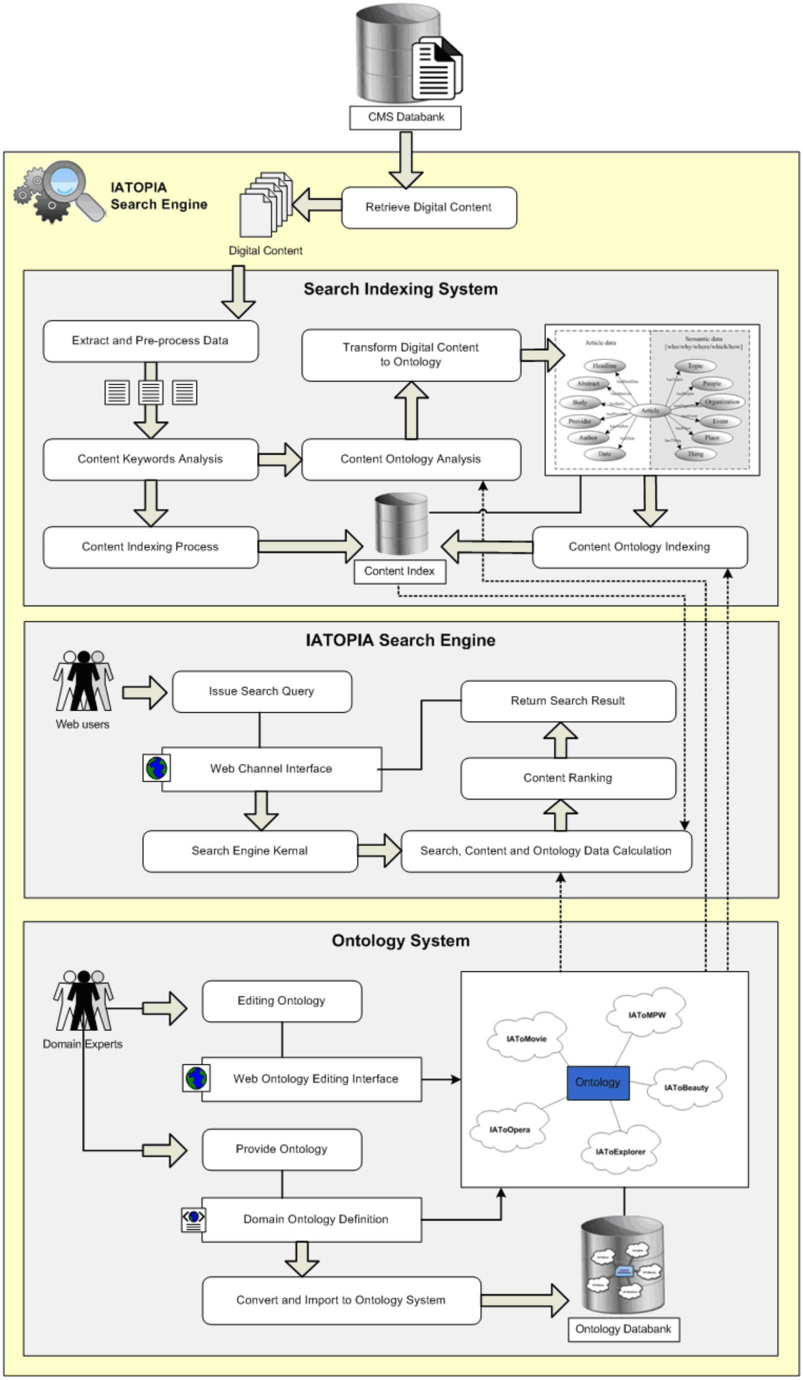


Fig. 10.2 Process flow of IATOPIA iCMS KnowledgeSeeker

10.1.3 Ontology System

IATOPIA Ontology System maintains and stores different ontology knowledge for different domains. It consists of an ontology databank which provides ontology data for the Search Indexing System to process content ontology analysis and provide for the Search Engine to process ontological querying and information search (see Chapter 3 for the ontology model defined in KnowledgeSeeker).

Process Description

1. The core ontology consists of a 10-domain-ontology (used in News Channel), which is generated by the ontology learning process and it is used for news article analysis. Some other ontologies include an opera ontology (used in Opera Channel) and a movie ontology (used in Movie Channel), they are maintained by a group of domain experts and it is mainly used for multimedia digital asset management.
2. The ontologies can be defined through two methods: 1. Editing through web interface, and 2. providing a fully structured ontology tree to IATOPIA.
3. By the first method, domain experts use a web ontology editing interface to create, modify the ontology online. The interface is linked up with the ontology server and databank, the ontology creation and modification are updated instantly on user editing.
4. By the second method, the content provider (domain expert) provides a fully structured ontology tree in a well defined format (e.g. XML or Excel). IATOPIA ontology system can convert and import it into the Ontology Databank.
5. Ontology data stored in the ontology databank provides rich knowledge about different domains for the IATOPIA search indexing system and search engine to process for ontological operation (analysis, indexing, and searching).

10.1.4 Search Indexing System

IATOPIA Search Indexing System maintains and creates all ontology based indexes of digital contents for search engine operations.

Process Description

1. The system retrieves all digital content from the iCMS Databank Cluster, and then extracts the structured content, including title, text, date and all related metadata, processes with data cleansing, pre-processing, and analysis.

2. The system analyzes the text content, including some textual analysis such as word segmentation, matching, counting frequency, measuring the ranking, etc. The content analysis adopts an ontology based approach, which requires the IATOPIA Ontology System to provide ontology data as the knowledge for analysis.
3. The content analysis result is converted into a data storage format (an ontology content index), and then stored persistently in the Content Index Databank.
4. The content analysis finally transforms the digital content from raw text into a structured ontology data representation.
5. The Ontology Content Index Databank finally stores the index of all digital contents and it is created for the search engine for content searching process.

10.1.5 IATOPIA Search Engine

IATOPIA Search Engine receives search request and query from various applications built on top of the IATOIPA iCMS KnowledgeSeeker, such as content retrieval system, content management system, and web channel etc. IATOPIA Search Engine does not only search information by keywords, it also processes query and analyzes content by ontology. The search engines make use of the Content Index Databank and Ontology Databank effectively to process user search request.

Process Description

1. The search engine first accepts user search request and query from web interface (either on content retrieval system, content management system or through browsing web channel), and then processes with the basic query processing, such as query segmentation.
2. The processed search query is transferred to IATOPIA search engine system. The system gathers all required information, such as the query itself, the required domain ontology (from ontology databank), the required search index (from content index), and the original article databank if necessary.
3. The gathered information is then processed with an ontological data calculation and similarity measurement (as described in Chapter 3.5).
4. All measurement in the content index has been done according to the search algorithm. A content ranking is done for making a list of desired results.
5. The search result is rearranged with all information, such as the original article/ e-publication data (e.g. text, related multimedia, page number, etc.), and then a result page created for user.

10.2 IATOPIA Digital Asset Management System (DAMS)

IATOIPA Digital Assets Management System (DAMS) is a web application developed for IATOIPA.com. The DAMS is integrated with the IATOIPA iCMS KnowledgeSeeker to provide a comprehensive and intelligent ontology-based management system for user to archive, manage and search for their large amount of digit asset files (multimedia content).

10.2.1 DAMS System Architecture Overview

IATOPIA DAMS provides a centralized content databank cluster to categorize, manage, store and retrieve different types of digital asset, e.g. news articles, photos, videos and audio data. With the integration of IATOIPA iCMS Knowledge-Seeker and DAMS, users can define their own ontology concepts to annotate (tagging) the attributes for all digital assets for ease of maintenance.

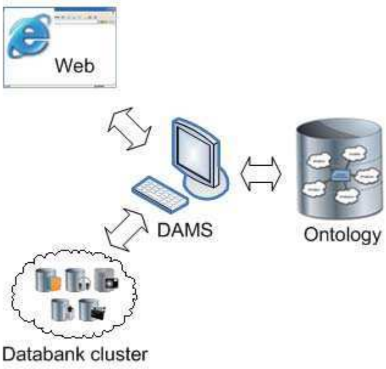


Fig. 10.3 DAMS architecture overview

10.2.2 IATOIPA iCMS Databank Cluster in DAMS

IATOPIA iCMS Databank cluster stores and integrates all digital asset files in DAMS. The databank also imports digital assets files from other iCMS module such as news collection module, e-Publication module, multimedia uploading module, article creation module, etc. The databank clusters (Figure 10.4) include:

- ***Text Data Cluster*** – store all text related contents, mainly including articles and all related information.

- ***Image Data Cluster*** – store all image related files, such as image from news articles, web channel, and file uploaded through the DAMS system.
- ***Video Data Cluster*** – store all video related contents, it consists of different video format for online video browsing, management, video editing, and video streaming.
- ***Audio Data Cluster*** – store all audio related contents, it consists of different audio format for online audio browsing, management, audio editing and audio, streaming service.
- ***E-Publication Data Cluster*** – store all e-publication contents, information, and files that are related and created from e-publication system.

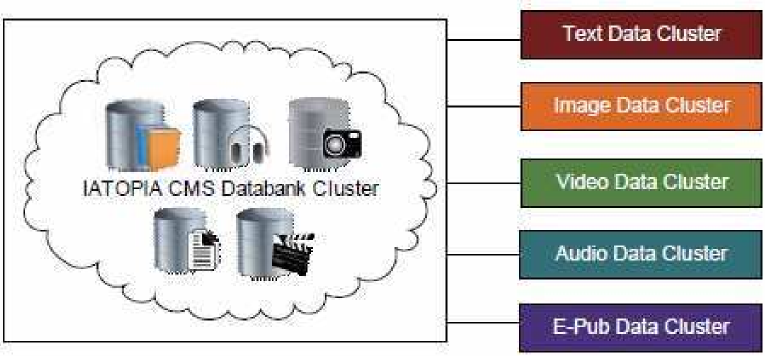


Fig. 10.4 IATOPIA iCMS databank cluster for DAMS

10.2.3 Ontology System in DAMS

IATOPIA iCMS Ontology System is the core technology to develop intelligent system module. Ontology is a computational knowledge model to conceptualize any object created in web channels. IATOPIA ontology module serves as core knowledge to associate with all conceptualized objects to create, manage and search for contents efficiently. The ontology system is also developed with automatic learning ability, i.e. upon creating web channels content, knowledge can be grown automatically. Figure 10.5 shows the web interface example of DAMS in movie domain, Figure 10.6 shows a Web Channel of movie domain which retrieves the content created in DAMS, and Figure 10.7 shows an ontology editing interface in DAMS for Chinese opera domain which is maintained by a group of Chinese Opera domain experts.

10.2.4 DAMS and Web Channel Interface Examples

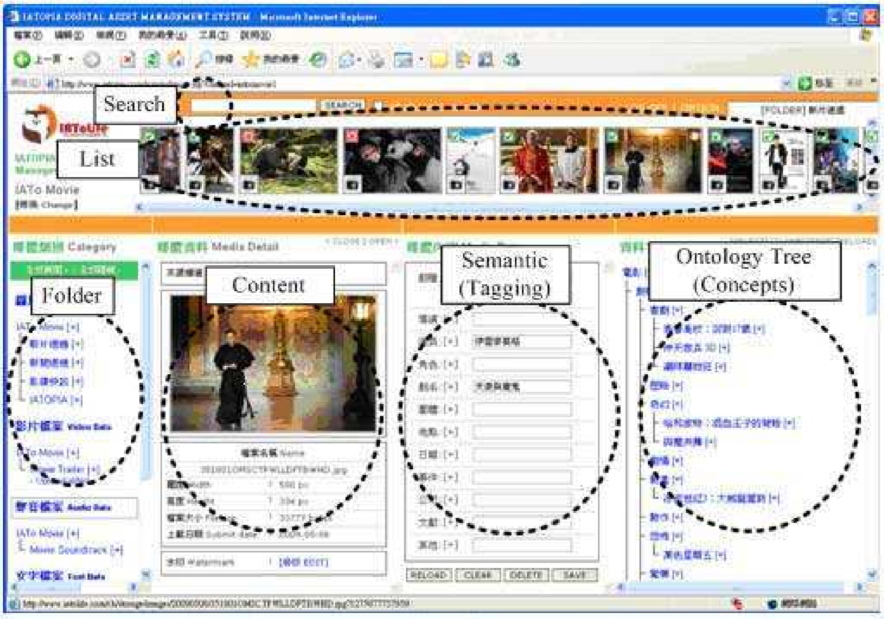


Fig. 10.5 DAMS interface for movie domain

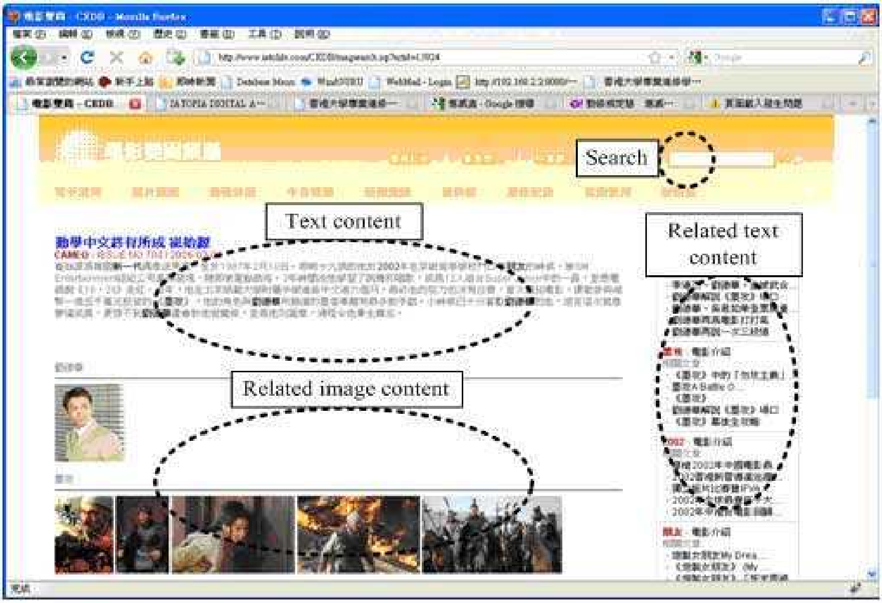
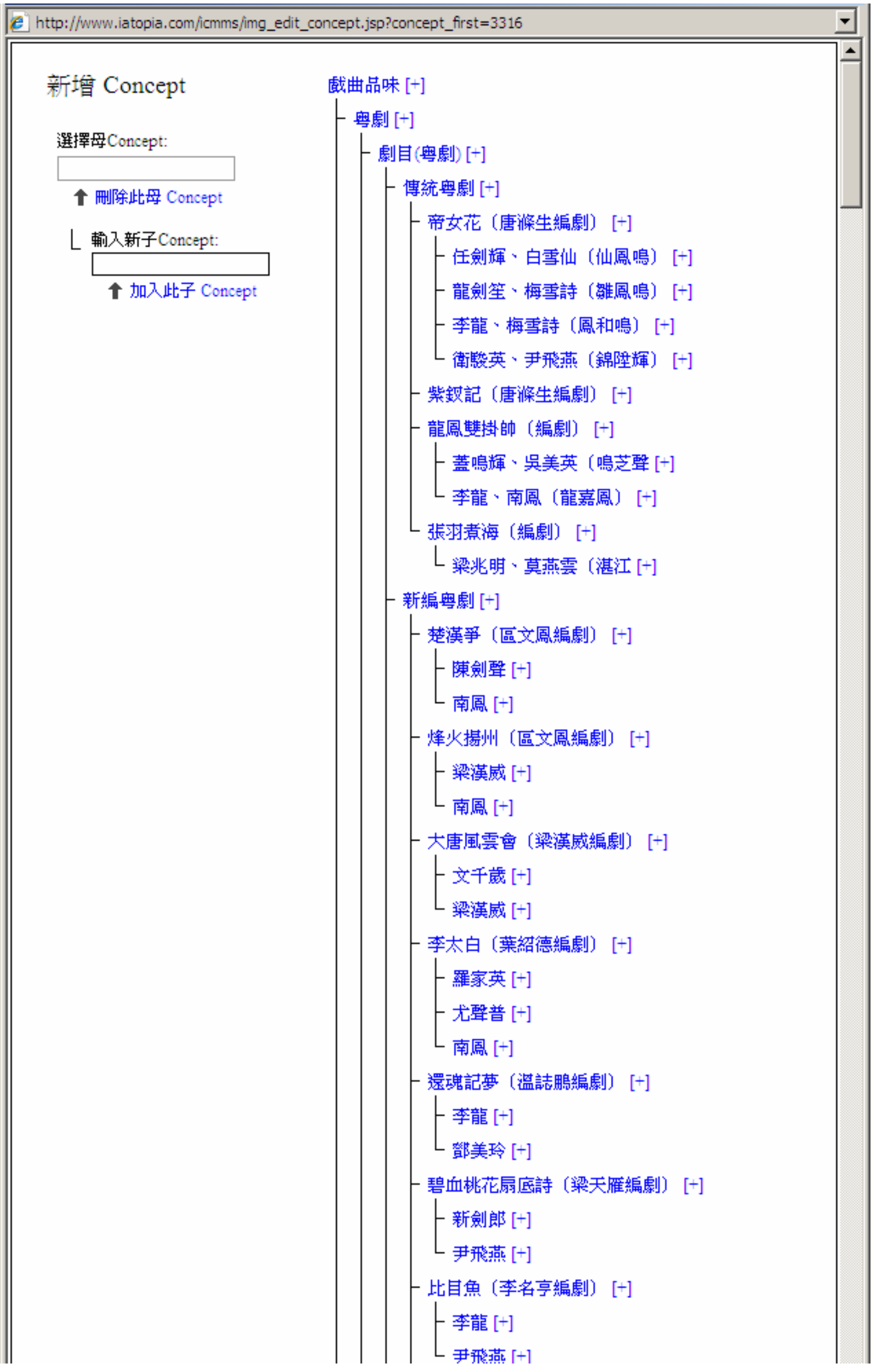


Fig. 10.6 Web channel linked to DAMS content for movie domain



- 戲曲品味 [+]
  - 粵劇 [+]
    - 劇目(粵劇) [+]
      - 傳統粵劇 [+]
        - 帝女花（唐滌生編劇） [+]
          - 任劍輝、白雪仙（仙鳳鳴） [+]
          - 龍劍笙、梅雪詩（雛鳳鳴） [+]
          - 李龍、梅雪詩（鳳和鳴） [+]
          - 衛駿英、尹飛燕（錦陞輝） [+]
        - 紫釵記（唐滌生編劇） [+]
        - 龍鳳雙掛帥（編劇） [+]
          - 蓋鳴輝、吳美英（鳴芝聲 [+]
          - 李龍、南鳳（龍嘉鳳） [+]
        - 張羽煮海（編劇） [+]
          - 梁兆明、莫燕雲（湛江 [+]
      - 新編粵劇 [+]
        - 楚漢爭（區文鳳編劇） [+]
          - 陳劍聲 [+]
          - 南鳳 [+]
        - 烽火揚州（區文鳳編劇） [+]
          - 梁漢威 [+]
          - 南鳳 [+]
        - 大唐風雲會（梁漢威編劇） [+]
          - 文千歲 [+]
          - 梁漢威 [+]
        - 李太白（葉紹德編劇） [+]
          - 羅家英 [+]
          - 尤聲普 [+]
          - 南鳳 [+]
        - 還魂記夢（溫誌鵬編劇） [+]
          - 李龍 [+]
          - 鄧美玲 [+]
        - 碧血桃花扇底詩（梁天雁編劇） [+]
          - 新劍郎 [+]
          - 尹飛燕 [+]
        - 比目魚（李名亨編劇） [+]
          - 李龍 [+]
          - 尹飛燕 [+]

Fig. 10.7 Example of Chinese Opera ontology tree maintained by domain experts.

Chapter 11
IATOPIA News Channel (IAToNews) – An Intelligent Ontological Agent-Based Web News Retrieval and Search System

Abstract. IATOPIA News Channel (IAToNews) is an online News Channel developed for IATOPIA.com. IAToNews is a web application integrated with the IATOPIA iCMS KnowledgeSeeker to provide a powerful and intelligent ontology-based information system for user to read and search news article (Chinese news) through a web browser. It consists of mainly an agent system for online news collection and an ontology system for news analysis and content personalization.

11.1 Introduction

IAToNews is a web platform for reading Chinese language RSS news feeds, and it has incorporated Intelligent Agent Technology (IAT) which will enable browsers to read RSS news articles and will also analyze all articles with further related articles provided to users automatically. In addition, it also allows each user to "build" their individual and personalized favorite news categories.

The source of these news feeds are obtained from official authority such as Radio Television Hong Kong, information Services Department of the Hong Kong Special Administrative Region Government, British Broadcasting Corporation together with other reputable sources such as Xinhuanet, Reuters, MSN, etc.

The main functions and features of IAToNews include:

- Intelligent Agent System for web news collection
- Ontology system for domain knowledge modeling
- 5D ontology system for news semantic analysis
- Personalized category

11.2 IAToNews System Architecture Overview

IAToNews system automatically collects the most updated news from different web sites of news providers (e.g. BBC Chinese, HK Government, RTHK, MSN,

E.H.Y. Lim, J.N.K. Liu, and R.S.T. Lee: Knowledge Seeker, ISRL 8, pp. 175 – 180.
springerlink.com

etc.). It carries out news integration tasks from a large amount of news source and deliver accurate and valuable information to users. It incorporates an ontology system for analyzing news contents and identifying the news topic automatically. The core ontology knowledge also enhances the news search engine so that it can provide more accurate and relevant results to users. Intelligence self-learning feature also provides news personalization to every registered user. Every user receives their personalized content based on their reading habit and interest. They can input their area of interest into the system or let the system learn it when they are reading news through the web site. So that users can receive contents that they are mostly interested and filtered out most of the uninterested contents.

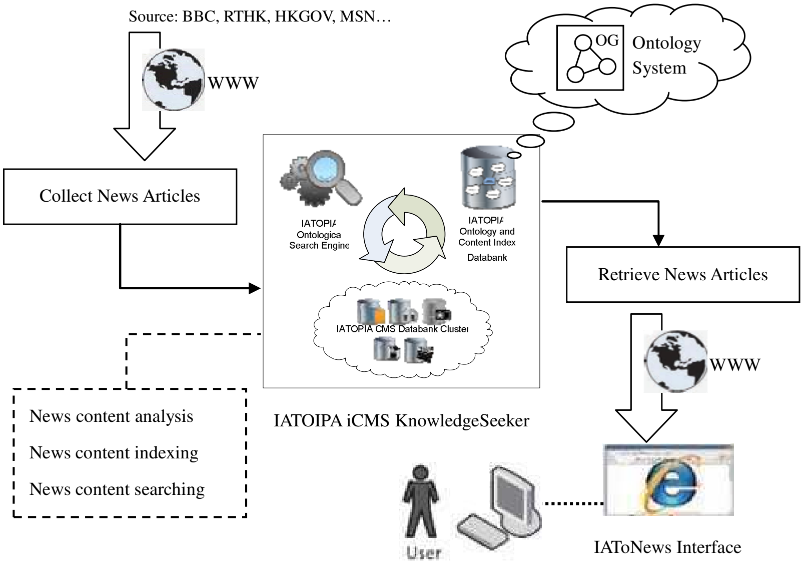

Source: BBC, RTHK, HKGOV, MSN…
WWW
OG
Ontology System
Collect News Articles
IATOPIA Ontologica Search Engine
IATOPIA Ontology and Content Index Databank
IATOPIA CMS Databank Cluster
Retrieve News Articles
WWW
News content analysis
News content indexing
News content searching
IATOIPA iCMS KnowledgeSeeker
User
IAToNews Interface

Fig. 11.1 IAToNews system architecture and information flow

11.3 Ontology System in IAToNews

The ontology system maintains and stores different ontology knowledge for different topics (domain). It consists of an ontology databank which provides ontology data for the search indexing system to process content ontology analysis and provides for the search engine to process ontological querying and content searching. There are two ontologies in IAToNews: 1. Article Ontology, 2. Topic Ontology (Domain Ontology), to be used for analyzing text document:

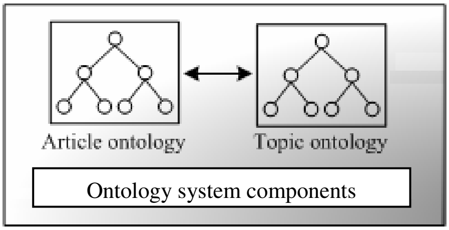


Fig. 11.2 Ontology system in IAToNews

11.3.1 Article Ontology

An ontology class "Article" is defined to describe the semantic content of a news article. The purpose of defining this ontology class is for the news annotation process. The Article ontology is separated with 2 types of data to store an article (Figure 11.3), the first one is simple article data (Table 11.1), and the second one is analyzed semantic data (Table 11.2).

Table 11.1 Article data in article ontology

Type	Description
Headline	The headline/title of the text
Abstract	The short abstract or short description of the entire text
Body	The main body and content of the text
Provider	The provider (source) of the content / article.
Author	Author who wrote the text
Date	Date of the text / article published.

Table 11.2 Semantic data in article ontology

Type	Description
Topic	The classified topic class of that article
People	The identified people, person included in the article.
Organization	The identified organization included in the article.
Event	The related and described events in the article.
Place	The place where the event occurred in the article.
Thing	All other things, object that are related in the article.

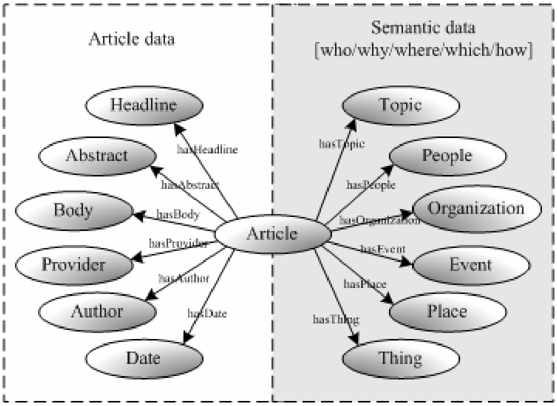


Fig. 11.3 Article ontology in IAToNews ontology system

11.3.2 Topic Ontology

The Topic Ontology class is defined for modeling the area of topic (domain of subject) in hierarchical relation, which is used to define the related topic of an article. The instances of topic are a set of controlled vocabularies for the ease of maintenance, sharing and exchange. There are 10 topics defined for news topic classification purpose as shown in Table 11.3. Every topic has their corresponding Domain Ontology Graph (*DOG*) representing the knowledge about the topic domain. It is used for ontology based content analysis (such as news topic identification), ontology-based content indexing and searching processes.

Table 11.3 Topics in IATOIPA KnowledgeSeeker news channel

Topic	Name	Name (English)
Topic 1	文藝	Arts and Entertainments
Topic 2	政治	Politics
Topic 3	交通	Traffic
Topic 4	教育	Education
Topic 5	環境	Environment
Topic 6	經濟	Economics
Topic 7	軍事	Military
Topic 8	醫療	Health and Medical
Topic 9	電腦	Computer and Information Technology
Topic 10	體育	Sports

11.3.3 Ontology Based Content Indexing

The indexing model converts all article contents which are originally stored in the iCMS content databank with an index structure. It is to extract and convert all content in text to the ontology format, and then store into content index databank. The types of ontology based article content index are shown in the Table 11.4. An article and its semantic entity content are stored into the IATOPIA ontology index databank for searching and retrieving.

Table 11.4 Types of ontology based content index of news article

Index name	Index type	Description
Topic	Domain Ontology Index	Domain Ontology Terms
People	Entity Ontology Index	Ontology Entity
Organization	Entity Ontology Index	Ontology Entity
Event	Entity Ontology Index	Ontology Entity
Place	Entity Ontology Index	Ontology Entity
Thing	Entity Ontology Index	Ontology Entity
Headline	String	Simple data type
Abstract	String	Simple data type
Body	Text	Simple data type
Provider	String	Simple data type
Author	String	Simple data type
Date	Date Time	Simple data type

11.4 IAToNews Web Interface Examples

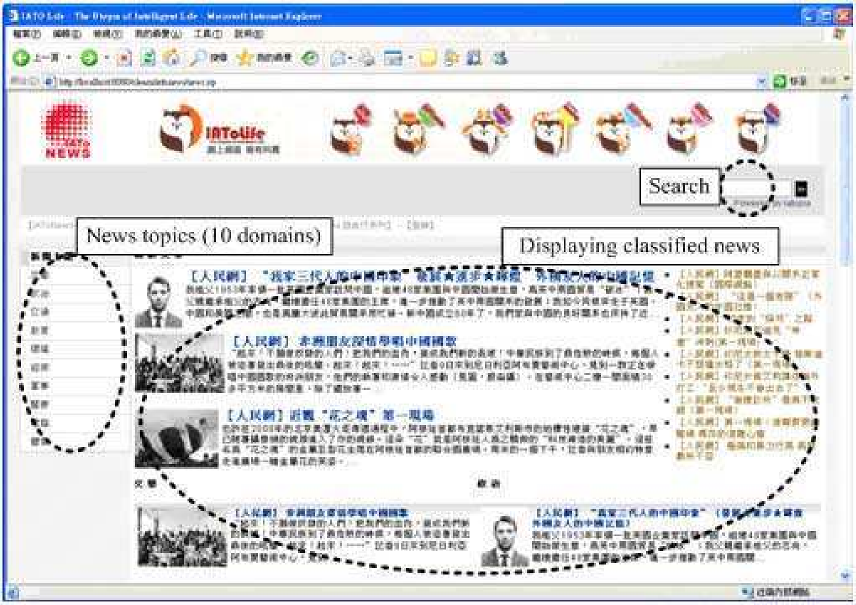


Fig. 11.4 Main page displaying classified news in IAToNews

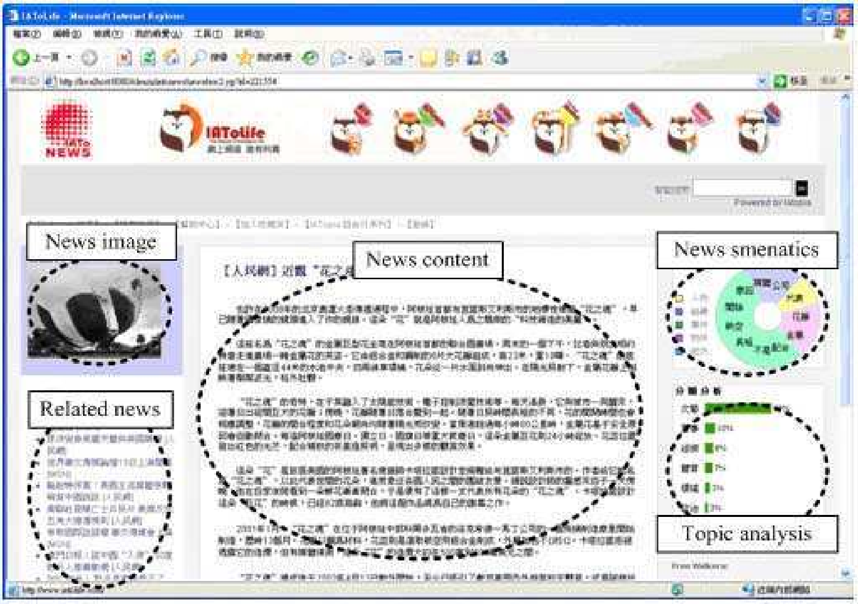


Fig. 11.5 Display news content and related info in IAToNews

Chapter 12
Collaborative Content and User-Based Web Ontology Learning System

Abstract. This chapter presents a Collaborative Ontology Learning Approach for the implementation of an Ontology-based Web Content Management System (OWCMS). The proposal system integrates two supervised learning approach - Content-based Learning and User-based Learning Approach. The Content-based Learning Approach applies text mining methods to extract ontology concepts, and to build an Ontology Graph (*OG*) through the automatic learning of web documents. The User-based Learning Approach applies features analysis methods to extract the subset of the Ontology Graphs, in order to build a personalized ontology. Intelligent agent approach is employed to capture user reading habit and preference through their semantic navigation and search over the ontology-based web content. This system combines the two methods to create collaborative ontology learning through an ontology matching and refinement process on the ontology created from content-based learning and user-based learning. The proposed method improves the validness of the classical ontology learning outcome by user-based learning refinement and validation.

12.1 Introduction

Nowadays, information, especially Web information is growing up at an exponential rate. In contrast, the information processing schemes become extremely difficult with a lot of manual intervention. Without a good solution to extract useful and meaningful information from raw data, such information "flooding" over the Internet becomes a disaster. For example, we need a lot of human resources to handle the data. This is very inefficient to prepare the information from raw information by hand. Moreover, like "the passion for love and hate" there has been a fine line between the "pure information" and "processed knowledge". The processed knowledge has lots of potential to gain advance in the search engine, products recommender system, etc.

Besides that, in Web 2.0, most of the users start to communicate with each other over the Internet. They use the Internet platform to provide knowledge for satisfying their needs. On the other hand, they will provide some keywords to search for

E.H.Y. Lim, J.N.K. Liu, and R.S.T. Lee: Knowledge Seeker, ISRL 8, pp. 181 – 194.
springerlink.com

their wants and needs. It gives a huge inducement to build a Collaborative Content and User-based Web Ontology Learning System to allow the Internet user to use a much more "knowledgeable" search engine with a personalized agent-based ontology as the kernel.

12.2 Background

12.2.1 Problem of Building a Generalized Ontology to Satisfy Daily Life

In Web Wide Web, it contains huge amount of information which likes a treasure. However, we cannot simply use such valuable information with any good "mining-tools". It is because we do not have general and satisfactory methods to retrieve and use the information. Although we found the generalized and satisfactory ontology to retrieve the information, everyone has their own interests and ideas. It is difficult (impossible) to satisfy the needs and wants of every Internet user with the same ontology, hence the search engine. With the popularity of Internet all over the world, it is a real need and temptation to create a framework for building a Web system with generalized ontology and to use it with personalization methods to solve this problem.

12.2.2 Semantic web

The current Web is largely built on HTML. HTML is originally designed for human consumption only. The problem of the current web architectures is that the Web systems are not designed to "understand" the Web content on their own. The Semantic Web is designed to solve this problem, by enriching web content with markup data. This markup data means to add more structural information to the semi-structured information in HTML page. This markup data gain benefits in machines understandability. Therefore it can enhance agent application to process web content. There is also close relationship between ontology and semantic web as ontology is the key element for building up semantic web content.

This section describes the Semantic Web architecture defined by W3C, which is about the underlying concepts and technologies supported for developing a semantic web. And then methods and process for semantic web development are discussed.

12.2.3 Web Channels

Web Channel Technology is coined by Dr. Raymond Lee in 2006 and served as part of the Web 3.0 Intelligent Agent-based Technology (IAT). It is an semantic web system which contains many general concepts with a set of specified domains. It includes a content management system, with intelligent search engine.

One important feature behind each web channel is that there have been strong and large domain experts to provide the knowledge contents. So it let us have enough and valuable data and information to perform a general ontology learning and validation.

12.2.4 BuBo (Feedback, Personalization with User Ontology)

BuBo is a collaborative platform / browser that provide feedback and personalization with ontology semantic web. It is an integration of Web Browser Technology, Ontology-based Search Engine and Intelligent Agent Technology. In the client's perspective, BuBo can provide more intelligent based service such as a more powerful ontology-based search engine and user-personalization services. In the backend, BuBo possesses an ontology-based content management agent, which link-up the knowledge-based of each designated Web Channel domain. Moreover, BuBo is able to collect the user's feedbacks, reading rabbits and perform an agent-based ontology semantic web browser with e-library. Download BuBo: http://www.iatolife.com/life/sw/bubo/

12.3 Methodology

This section describes the details of the proposed Collaborative Ontology Learning Approach, and the implementation methods to create the Ontology-based Web Content Management System. Section 12.3.1 describes the system architecture, the different stages and processes of the ontology learning system. Section 12.3.2 describes the process of the Content-based Ontology Learning Process. Section 12.3.3 describes the process of User-based Ontology Learning Process. Section 12.4 describes the collaborative approach to refine and improve the validness of the ontology learning outcome. Section E describes the web ontology application implemented in web channels.

12.3.1 Overview of System Architecture

The collaborative learning approach is divided into mainly two processes – Content-based Ontology Learning Process and User-based Ontology Personalization Learning Process. Figure 12.1 shows the system architecture of the learning approach.

The figure shows the overviews of data flow in the entire learning system. Two learning processes are basically processed separately, in that the learning outcome from Content-based learning process will be created as the basic input of User-based Learning Process. The ontology refinement and validation are taken place in user-side, which modify and refine the ontology and input back to the Content-based Learning Process to create a complete learning cycle to improve the validness of the Ontology.

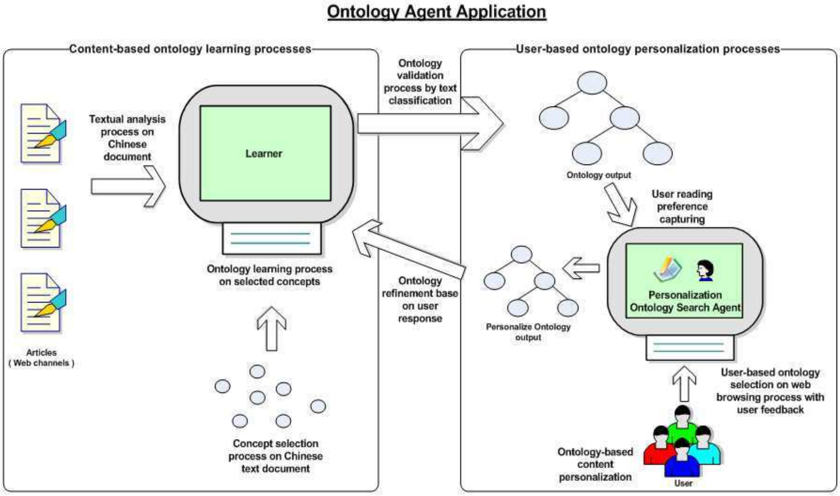

Ontology Agent Application
Content-based ontology learning processes
User-based ontology personalization processes
Textual analysis process on Chinese document
Learner
Ontology validation process by text classification
Ontology output
User reading preference capturing
Ontology learning process on selected concepts
Ontology refinement base on user response
Personalization Ontology Search Agent
Personalize Ontology output
Articles (Web channels)
User-based ontology selection on web browsing process with user feedback
Concept selection process on Chinese text document
Ontology-based content personalization
User

Fig. 12.1 Ontology agent application

12.3.2 Content-Based Ontology Learning Process

This Content-based Ontology Learning Process is comprised with four main steps as shown in Figure 12.2. They are 1. Textual Analysis, 2. Concept Selection, 3. Ontology Learning, and 4. Ontology Validation.

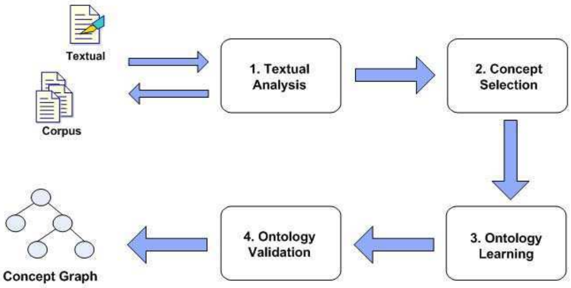


Fig. 12.2 Four content-based ontology learning processes

The ontology learning outcome – Ontology Graph (OG) is defined in this learning process. In the representation of Ontology Graph in Figure 12.3, we define different types of knowledge units according to their level of complexity to comprise knowledge. A knowledge unit is any objects in the Ontology Graph that give semantics expression:

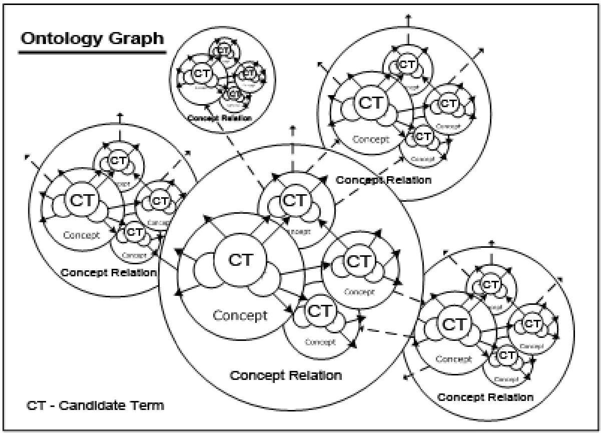


Fig. 12.3 Ontology Graph

1. Candidate Term (CT) – the smallest units that extracted in the form of a sequence of Chinese characters, those are meaningful words in human perspective.
2. Concept (C) – one or more candidate terms groups together with explicit relations to other knowledge unit, it is the basic knowledge unit in the ontology graph.
3. Concept Relation (CR) – the weight direct relations between two concepts. That defines how two concepts relate to each other.
4. Ontology Graph (OG) – The entire knowledge unit created by groups of concepts, representing a comprehensive knowledge of the domain of a web channel.

1) Textual analysis process on Chinese document

Textual analysis process on Chinese document in the web channels requires a list of common Chinese terms and special terminology of each web channels domain. Common Chinese terms are extracted from an electronic dictionary, such as HowNet, containing over 50000 distinct Chinese words, are used as the initial term list for textual analysis process. Special terminology of web channels domain, such as named entity, product brands, product model, etc. are human defined. Special terminology combined with the initial term list is the only predefined knowledge in the ontology learning process. A maximal matching algorithm is then applied to the term list and the web channels document to extract a list of candidate terms (CT), such that every term in the list exists at least once within all web channels.

2) Concept selection process on Chinese text document

Candidate terms (CT) contains no relations to web document, we define a term as Concepts (C) if the term contains a weighted relation to a domain. Thus, the Concept selection process aimed to select a set of Concepts that is related to a web channels, such that every web channel is linked to their related Concept through a weighted Concept Relation (CR). This process is done by a statistical term-to-class independence test. A term *t* refers to every CT and a class *c* refers a web channel (a specific domain). The independence value is calculated through a chi-square statistical measurement, as expressed in the 2-way contingency table. $O_{t,c}$ $(O_{\neg t,\ c})$ is the observed frequency of a term that occurs (or not occurs) in a web channel $O_{t,\neg c}$ $(O_{\neg t,\neg c})$ is the observed frequency of a term that occurs (or not occurs) in other web channels. Compared to the expected frequency $E_{i,j}$ where $i \in \{t, \neg t\}$ and $j \in \{c, \neg c\}$. $E_{i,j}$ is defined as:

$$E_{i,j} = \frac{\sum_{a\in\{t,\neg t\}} O_{a,j} \sum_{b\in\{c,\neg c\}} O_{i,b}}{N}$$

Chi-square statistics for term t and class c is defined as:

$$\chi^2_{t,c} = \sum_{i\in\{t,\neg t\}} \sum_{j\in\{c\neg c\}} \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}}$$

3) Ontology learning process on selected concepts

The concept selection measure dependency between a term and a class, but it does not measure the relation between every term inside the class. Therefore, a further measurement of the concept relation between every concept in the class is required, and is known as Ontology learning process. This measurement applies similar chi-square statistical term-to-term independency test. Equation is changed as to measure the chi-square value for a term t_a and another term t_b

$$\chi^2_{t_a,t_b} = \sum_{i\in\{t_a,\neg t_a\}} \sum_{j\in\{t_b,\neg t_b\}} \frac{(O_{i,j} - E_{i,j})^2}{E_{i,j}}$$

Ontology Graph is created by selecting a certain highest weighted Concepts inside a class (web channels). The vector of the dependency values between every term within a class is converted into a directed graph $G = (V, A)$ where V is the set of selected Concepts (C) $V = \{t_1, t_2,\ldots,t_{k-1}, t_k\}$ and A is the set of directed and weighted Concept Relation (CR), $A = \{(t_1,t_1, \chi^2_{t_1,t_1}), (t_1,t_2, \chi^2_{t_1,t2}),\ldots, (t_k,t_{k-1}, \chi^2_{t_k,t_{k-1}}), (t_k,t_k, \chi^2_{t_k,t_k})\}$. A threshold *k* is selected to create an Ontology graph containing *k* number of Concepts (Figure 12.4).

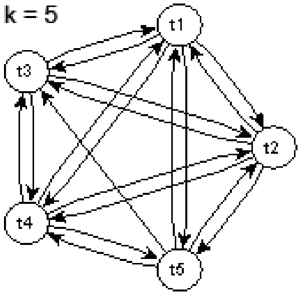


Fig. 12.4 Ontology graph containing *k* number of Concepts

4) Ontology validation process by text classification

Measurement of the validness of an Ontology Graph is done by a text classification process. This process aims to validate how good of an Ontology Graph can represent a class (Web channel). Since the number of Concepts for a class is very large, we also need to evaluate the threshold *k* to be selected for creating an Ontology Graph with an optimal size. In the text classification model, every web channel document is represented by a vector space model. Each document is represented by a term-frequency vector $TF =< tf_1, tf_2, ..., tf_n >$, and the term from the document is extracted by the matching algorithm with the candidate term list. A score vector is calculated for document *d* for every term t_i, score function s_i for a term t_i is defined as:

$$s_{t_i} = \begin{cases} \chi^2_{t_i,c} & if \quad t_i \quad exist \quad in \quad d \\ 0 & \qquad else \end{cases}$$

12.3.3 User-Based ontology Personalization Process

In this process, the User-based Ontology Personalization relies on the periodic knowledge, which is learned by the Ontology Learner (Ontology Graph, OC).

The OG changing and giving the personalization OG output will accompany with the user reading preferences and each user's Personalization Ontology Search Agent learning result. The following sections will describe the whole process of the framework.

1) BuBo (e-library, web browsing application)

Bubo is a web browsing application with e-library system. In the e-library system, user can subscribe the e-magazine. E-magazine has different categories, e.g. leisure, travel, news, technology. Some of magazines may belong to web channel. For example, Hong Kong Beauty belongs to IAToBeauty.com. In each web channel, it will have its different categories, e.g. Skin Care, Make Up, Nail DIY.

2) User reading preference capturing

The user uses the BuBo to browse the web channel, e-Mag, and e-Book. In the BuBo, there is built-in capture user preferences system. This system is based on the user reading habit to capture the preferences. The user reading habit will be stored at the XML in his/her PC, and provide the input for the user's Personalization Ontology Search Agent.

The user preferences XML stored the user preference information, each user has his own XML. This XML will be passed to the user's Personalization Ontology Search agent for performing the personalization. Figure 5. shows the structure of the user preferences XML.

```
<user id = 3569>
            <type id = 1 category = 1 page = 10>
                        <Concept id = 10 />
            <type />
            <type id = 2 page = 101 />
            <type id = 3 page = 78 />
            <type id = 4 concept =”Andy Lau” />
<user />
```

Fig. 12.5 Structure of the user preferences XML

"User id" is used to define which user's reading preferences. "Type" is the type of the media. In the system, there has 4 types of media for user reading.

1. Web channel
   a. The system will capture the user reading pages and which category the user is reading.
   b. If there has the concept appeared in the type 1, and the user uses the concept cross linkage for further reading then in <type id =1 page =XXX> there will have a concept.
2. e-Mag
   a. The user read the e-Mag with the specific page.
3. e-Book
   a. The user read the e-Book with the specific page.
4. Search and customized concept

3) Ontology- base content personalization

User's Personalization Ontology Search Agent

User's Personalization Ontology Search Agent responds to learn user preferences, collect the user feedback, assist the user to search their interests, and report learning result to Ontology learner for further learning.

First of all, it needs to connect to the Ontology Learner for helping the user retrieve the periodic knowledge and use the periodic knowledge (*OG*) to initiate the personalization search engine. The User's Personalization Ontology Search Agent will base on the user preference and the user feedback to adjust and customize the user's own search engine and semantic web. Figure 12.6 shows the top level (OG) and Figure 12.7 shows the example structure of the (*OG*).

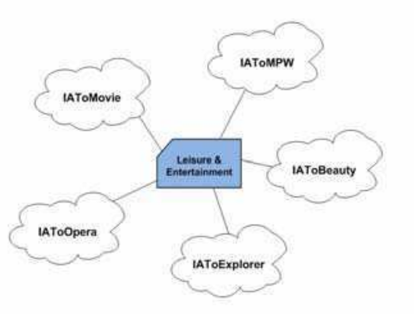


Fig. 12.6 The top level Ontology Graph

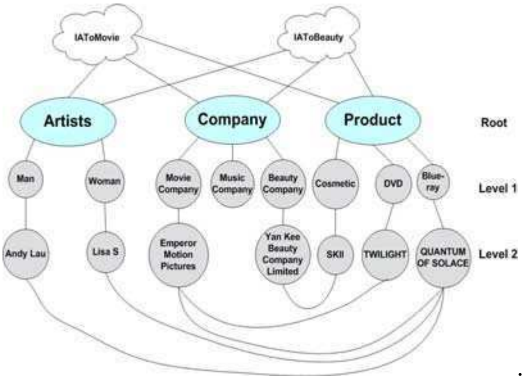


Fig. 12.7 The example structure of Ontology Graph

There are three cases for the User's Personalization Ontology Search Agent to trigger the personalized ontology graph (POG). 1. User search from the web channels or referral link to perform cross search 2. User browses the web channels, e-Mag, or e-Book, 3. User customized his/her search engine.

User searches from the web channels or referral link to perform cross search. When the user is searching from the web channels, the user must provide some keywords γ_i. Then the User's Personalization Ontology Search Agent bases on γ_i to search from the POG.

The user searches form the web channels:

1. POG contains

- If γ_i is the subset of the POG, then the User's Personalization Ontology Search Agent will base on the POG and update the semantic web cross linkage result.
- In the same case, if user uses the referral links to perform cross search, then γ_i's $i = 1$

2. POG does not contain

- If γ_i is not a subset of the POG, then User's Personalization Ontology Search Agent will communicate with Ontology Learner, and try to search for the periodic knowledge (OG).
- If γ_i is a subset of the OG, then User's Personalization Ontology Search Agent will retrieve 2 levels of (OG) and update the semantic web which is based on the search result (sub-graph of the OG). After that the User's Personalization Ontology Search Agent will retrieve the sub-graph sg from (OG). The sub-graph will retrieve from the category of γ_i to γ_i's stays level + 1 level for renewing the POG. In this case, there may have $[\gamma_1 ... \gamma_n]$ keywords, then the result will be based on the number of γ_i to retrieve the $[sg_1 ... sg_n]$ (The number of sg $\leq$ the number of γ).

User browses the web channels, e-Mag, or e-Book

When the user uses the BuBo to browse the web channels, e-Mag, or e-Book, the capture user preferences system will capture what the user has been browsing and record it. For the web channels, it will capture categories and pages that the user has browsed. And for the e-Mag and e-Book cases, it just captures which page is browsed by the user.

Each page contains many concepts (C), it is a subset of (C) $V = \{t_1, t_2, \ldots, t_{k-1}, t_k\}$ and there must contain a subset of directed and weighted Concept Relation (CR), $A = \{(t_1, t_1, \chi^2_{t_1,t_1}), (t_1, t_2, \chi^2_{t_1,t2}), \ldots, (t_k, t_{k-1}, \chi^2_{t_k,t_{k-1}}), (t_k, t_k, \chi^2_{t_k,t_k})\}$, that means each page has a sub-graph (SG).

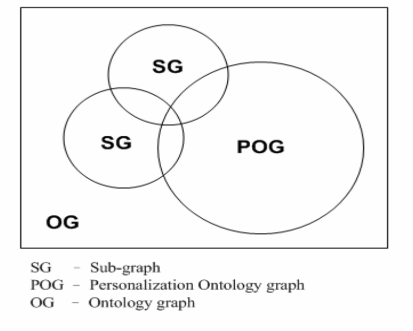


Fig. 12.8 SG interception with the POG and forming new POG

If (SG) intercepts with the (POG), then the new (POG) is formed by the User's Personalization Ontology Search Agent. Moreover, if (SG) "S1" intercepts with the other (SG) "S2", and "S2" intercepts with (POG), then $\{S1, S2\} \in POG$. Figure 12.8 shows (SG) interception with the (POG) and forms new (POG)

User customized his/her search engine

In this system, user can use BuBo which provide interface to insert the concept into his/her User's Personalization Ontology Search Agent to enhance and find touch the POG. After that User's Personalization Ontology Search Agent will perform the recalculations which like the situation in User search from the web channels.

The user can use the BuBo to insert concepts for each web channel. Then the concepts plugged into the (POG) by Personalization Ontology Search Agent. After that the user can have additional recommendation based on their customized concepts. In the same cases, if the user has subscribed the e-Mag or e-Book, then the (POG) is updated by Personalization Ontology Search Agent and let user search though the e-Mag and e-Book which are in their own personalized e-library. Figure 12.9 shows the process of customization and how the user uses the BuBo to interact with the User's Personalization Ontology Search Agent.

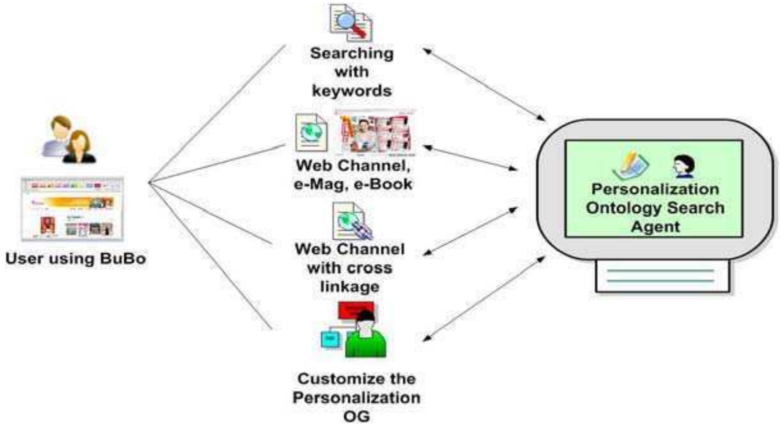

Searching with keywords
Web Channel, e-Mag, e-Book
Web Channel with cross linkage
Customize the Personalization OG
User using BuBo
Personalization Ontology Search Agent

Fig. 12.9 The process of customization and how the user uses the BuBo to interact with the User's Personalization Ontology Search Agent

Ontology refinement base on user response

After each client has its own POG, then the User's Personalization Ontology Search Agent can send back the POG to refine the weight of the relation in the OG. It will clarify the growth of the POG and process which is the updated knowledge and collaborate with the Ontology learner to adjust the periodic knowledge.

12.4 Implementation

12.4.1 Architecture of the Collaborative System Implementation

The collaborative system is divided into two parts. One part is the client application BuBo with User's Preferences Search Agent, the other part is the ontology learning server and web channel server as shown in Figure 12.10.

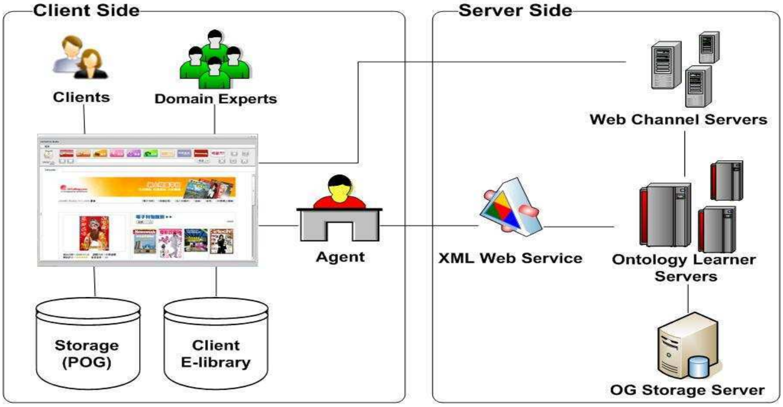


Fig. 12.10 The ontology learning server and web channel server

In the client side, there are two main types of user, general clients and domain experts. General Clients use the BuBo with User's Preferences Ontology Search Agent to browse the web channel or other web page. Also they can use the BuBo to access their e-library for reading their e-magazines or e-books. BuBo with User's Preferences Ontology Search Agent provides the personalization ontology search service. The ontology search agent helps clients to search their (POG) and update the web channel's recommendation linkages, which let user perform the cross searching and have a professional search recommendation.

On the other hand, the Domain Experts use BuBo as an ontology editor. They define the general ontology in BuBo, and the BuBo uses the XML Web Service to communicate with the Ontology Learner Servers to edit the (OG).

In the server side, there has three components, Web channel Servers, Ontology Learner Servers and (OG) storage servers. Web channel Servers are the semantic web servers which provide the semantic web hosting services. Then the Ontology Learner Servers are responding to learn and refine the (OG). Finally, the (OG) will be stored in the (OG) storage servers. Figure 12.11 shows the BuBo Structure.

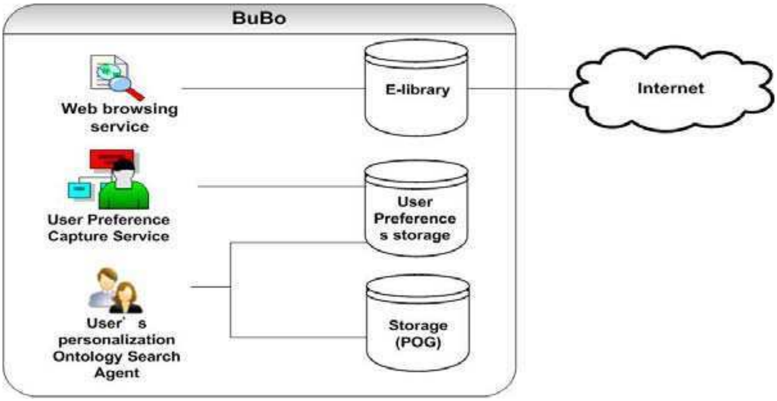


Fig. 12.11 BuBo Structure and responsibilities

12.4.2 Structure of the Specified Domains Ontology with Generalized OG

Different IAToLife Web Channel corresponds to different specified domain ontology with generalized ontology OG, e.g. IAToMovie, IAToNews, IATo-Beauty, etc. and shows different topic-based generalized ontology graph, and each topic-base generalized OG may have relation to make a big generalized OG. Figure 12.12 shows the example of the topic-base generalized OG. The example contains two parts, red one is in IAToMovie domain, and yellow is in IAToBeauty domain. The root node is in purple. However, some of them have interception which different topic-base generalized OG may share the same concept with the relation, in Figure 12.12 with green one's concepts that represent the situation.

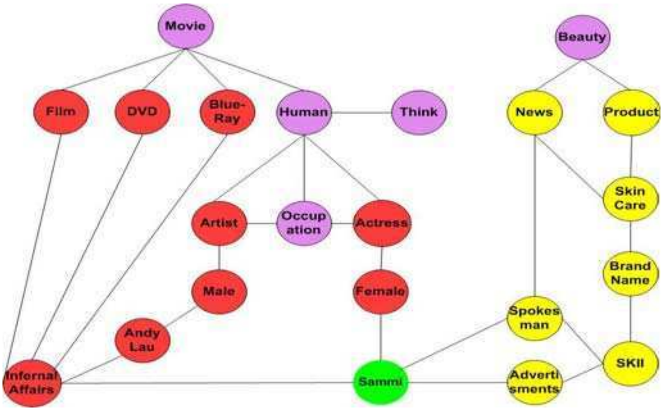


Fig. 12.12 The example of the topic-base generalized OG

12.4.3 Ontology-Based Search Engine within BuBo

User can search their interest from the BuBo. First, the User's Personalization Ontology Search Agent searches through the (POG). It then gives the recommendation from topic-base generalized (POG). E.g. if the user searches for "Sammi" (the green node in Figure. 11), then the agent will extract the topic base first such that if the client is browsing IAToBeauty, then the agent will give the yellow's nodes for the top results, and then it will base on the (POG) to give one level of red node e.g. "Infernal Affairs" is another recommendation result for another web channel.

12.5 Conclusions

In this chapter, the Collaborative Content and User-based Web Ontology Learning System had been development. This involved a general semi-automation ontology learning framework with user personalization.

The general ontology graph has been built and each user uses the BuBo to provide the reading habit, and the User Personalization Ontology Search Agent helps the user to make a personalization search engine, and communicate with the Ontology Learner to suggest and reformat the Ontology Graph for enhancing the search engine of web channels. The solution has been developed and the OG can work for much more advanced search engine or other application.

References

Abdoullaev, A.: Reality Universal Ontology and Knowledge Systems. Toward the Intelligent World. IGI Publishing, New York (2008)

Adomavicius, G., Tuzhilin, A.: Toward the Next Generation of Recommender Systems: a Survey of the State-of-the-art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering 17(6), 734–749 (2005)

Aggarwal, C.C., Gates, S.C., Yu, P.S.: On Using Partial Supervision for Text Categorization. IEEE Transactions on Knowledge and Data Engineering 16(2), 245–255 (2004)

Agichtein, Y., Gravano, S.: Snowball: Extracting Relations from Large Plain-text Collections. In: Proceedings of the Fifth ACM International Conference of Digital Libraries, pp. 85–94 (2000)

Alani, H.: Ontology Construction from Online Ontologies. In: Proceedings of the 15th World Wide Web Conference, pp. 491–495 (2006)

Anderberg, M.R.: Cluster Analysis for Applications. Academic Publishers, New York (1973)

Angelaccio, M., Catarci, T., Santucci, G.: Qbd: a Graphical Query Language With Recursion. IEEE Transactions on Software Engineering 16(10), 1150–1163 (1990)

Anick, P.G.: Integrating Natural Language Processing and Information Retrieval in a Troubleshooting Help Desk. IEEE Expert: Intelligent Systems and Their Applications 8(6), 9–17 (1993)

Aristotle: Metaphysics. In: Adler, M.J. (ed.) Great Books of the Western World, vol. 1, Encyclopedia Britannica, Chicago (1990)

Arpirez, J.: Reference Ontology and (onto)2agent: the Ontology Yellow Pages. Knowledge and Information Systems 2(4), 387–412 (2000)

Artale, A., Franconi, E., Guarino, N., Pazzi, L.: Part-whole Relations in Object-centered Systems: an Overview. Data and Knowledge Engineering 20(3), 347–383 (1996)

Avesani, P., Giunchiglia, F., Yatskevich, M.: A Large Scale Taxonomy Mapping Evaluation. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 67–81. Springer, Heidelberg (2005)

Barbara, D., Molina, H.G., Porter, D.: A Probabilistic Relational Data Model. In: Proceedings of the International Conference on Extending Database Technology on Advances in Database Technology, Venice, Italy, pp. 60–74 (March 1990)

Barr, A., Davidson, J.: Representation of Knowledge. In: Barr, A., Feigenbaum, E. (eds.) Handbook of Artificial Intelligence, Stanford University, Computer Science Dept. Report No. STAN-CS-80-793 (1980)

Berendt, B., Hotho, A., Stumme, G.: Towards semantic web mining. In: Horrocks, I., Hendler, J. (eds.) ISWC 2002. LNCS, vol. 2342, pp. 264–278. Springer, Heidelberg (2002)

Berners-Lee, T.: Notation 3: an Rdf Language for the Semantic Web. Tech. Rep. World Wide Web Consortium, w3c (2000)

Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American, 1–19 (2001)

Bhogal, J., Macfarlane, A., Smith, P.: A Review of Ontology Based Query Expansion. Information Processing and Management: an International Journal 43(4), 866–886 (2007)

Blomqvist, E.: Fully Automatic Construction of Enterprise Ontologies Using Design Patterns: Initial Method and First Experiences. In: Proceedings of the 5th International Conference on Ontologies Databases and Applications of Semantics (2005)

Blomqvist, E.: Semi-automatic Ontology Engineering Using Patterns. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 911–915. Springer, Heidelberg (2007)

Blondel, V.D., Gajardo, A., Heymans, M., Senellart, P., Dooren, P.V.: A Measure of Similarity Between Graph Vertices. Corr. (2004)

Bonino, D., Corno, F., Pescarmona, F.: Automatic Learning of Text-to-concept Mappings Exploiting WordNet-like Lexical Networks. In: Proceedings of the ACM Symposium on Applied Computing, Santa Fe, New, Mexico, March 13-17, pp. 1639–1644 (2005)

Bontas, E., Mochol, M., Tolksdorf, R.: Case Studies on Ontology Reuse. In: Proceedings of the 5th International Conference on Knowledge Management (i-know 2005), Graz, Austria, pp. 345–353 (2005)

Borst, W.N., Akkermans, J.M., Top, J.L.: Engineering Ontologies. International Journal on Human Computer Studies 46(2), 365–406 (1997)

Boulton, M.: Icons Symbols and a Semiotic Web (2005), Rerieved from `http://www.markboulton.co.uk/journal/comments/some-thoughts-about-signs`

Brachman, R., Anand, T.: The Process of Knowledge Discovery in Databases. In: Brachman, R., Anand, T. (eds.) Advances in Knowledge Discovery and Data Mining, pp. 37–57. AAAI Press, Menlo Park (1996)

Brank, J., Grobelnik, M., Mladenic, D.: A Survey of Ontology Evaluation Techniques. In: Proceedings of the 8th International Multi-conference Information Society Is 2005, Ljubljana, Slovenia (2005)

Brunzel, M.: The Xtreem Methods for Ontology Learning from Web Documents. In: Proceeding of the 2008 Conference on Ontology Learning and Population: Bridging the Gap Between Text and Knowledge, June 16, pp. 3–26 (2008)

Buitelaar, P., Ciomiano, P.: Ontology Learning and Population: Bridging the Gap Between Text and Knowledge. IOS Press, Amsterdam (2008)

Buitelaar, P., Cimiano, P., Magnini, B.: Ontology Learning from Text: Methods Evaluation and Applications. IOS Press, Amsterdam (2005)

Buitelaar, P., Olejnik, D., Sintek, M.: A Protégé plug-in for Ontology Extraction from Text. In: Bussler, C.J., Davies, J., Fensel, D., Studer, R. (eds.) ESWS 2004. LNCS, vol. 3053, pp. 31–44. Springer, Heidelberg (2004)

Buitelaar, P., Olejnik, D., Sintek, M.: Protégé Plug-in for Ontology Extraction from Text Based on Linguistic Analysis. In: Bussler, C.J., Davies, J., Fensel, D., Studer, R. (eds.) ESWS 2004. LNCS, vol. 3053, pp. 31–44. Springer, Heidelberg (2004)

Busagala, L.S.P., Ohyama, W., Wakabayashi, T., Kimura, F.: Improving Automatic Text Classification by Integrated Feature Analysis. IEICE - Transactions on Information and Systems E91-D(4), 1101–1109 (2008)

Cao, G., Nie, J., Bai, J.: Integrating Word Relationships Into Language Models. In: Proceedings of the 28th Annual International ACM Sigir Conference on Research and Development in Information Retrieval, pp. 298–305 (2005)

Cao, Y., Li, H.: Base Noun Phrase Translation Using Web Data and the Em Algorithm. In: Proceedings of the 19th International Conference on Computational Linguistics, Taipei, Taiwan, September 24 -August 1, pp. 1–7 (2002)

Castells, P., Fernandez, M., Vallet, D.: An Adaptation of the Vector-space Model for Ontology-based Information Retrieval. IEEE Transactions on Knowledge and Data Engineering 19(2), 261–262 (2007)

Catalano, C.E., Giannini, F., Monti, M., Ucelli, G.: A Framework for the Automatic Annotation of Car Aesthetics. AI Edam 21(1), 73–90 (2007)

Chan, C.W.: Knowledge Acquisition by Conceptual Modeling. Applied Mathematics Letters Journal 3, 7–12 (1992)

Chandler, D.: Semiotics for Beginners (2010), Retrieved from
`http://www.aber.ac.uk/media/Documents/S4B/sem02.html`

Chandrasekaran, B., Josephson, J.R., Benjamins, V.R.: What Are Ontologies and Why Do We Need Them? IEEE Intelligent Systems 14(1), 20–26 (1999)

Chen, H.: Machine Learning Approach to Document Retrieval: an Overview and an Experiment. Technical Report. University of Arizona MIS Department, Tucson AZ USA (1996)

Chen, L.L., Chan, C.W.: Ontology Construction from Knowledge Acquisition. In: Proceedings of Pacific Knowledge Acquisition Workshop (PKAW 2000), Sydney, Australia, December 11-13 (2000)

Cimiano, P., Völker, J.: Text2onto: a Framework for Ontology Learning and Data-driven Change Discovery. In: Montoyo, A., Muńoz, R., Métais, E. (eds.) NLDB 2005. LNCS, vol. 3513, pp. 227–238. Springer, Heidelberg (2005)

Cimiano, P., Völker, J.: Text2Onto. In: Montoyo, A., Muńoz, R., Métais, E. (eds.) NLDB 2005. LNCS, vol. 3513, pp. 227–238. Springer, Heidelberg (2005)

Cimiano, P., Hotho, A., Staab, S.: Comparing Conceptual Divisive and Agglomerative Clustering for Learning Taxonomies from Text. In: Proceedings of the 16th European Conference on Artificial Intelligence, Valencia, Spain, August 22-27, pp. 435–439 (2004)

Cimiano, P., Hotho, A., Staab, S.: Learning Concept Hierarchies from Text Corpora Using Formal Concept Analysis. Jair - Journal of AI Research 24, 305–339 (2005)

Cimiano, P., Hotho, A., Staab, S.: Learning Concept Hierarchies from Text Corpora Using Formal Concept Analysis. Journal of Artificial Intelligence Research 24, 305–339 (2005)

Cimiano, P., Hotho, A., Staab, S.: Learning Concept Hierarchies from Text Corpora Using Formal Concept Analysis. Journal of Artificial Intelligence 24, 305–339 (2005)

Clariana, R.B., Koul, R.: A Computer-based Approach for Translating Text Into Concept Map-like Representations. In: Proceedings of the First International Conference on Concept Mapping, pp. 125–133 (2004)

Clifton, C., Cooley, R., Rennie, J.: Data Mining for Topic Identification in a Text Corpus. IEEE Transactions on Knowledge and Data Engineering 16(8), 949–964 (2004)

Colace, C., Santo, M.D., Vento, M.: An Automatic Algorithm for Building Ontologies from Data. In: Proceedings of the International Conference on Information and Communication Technologies: from Theory to Applications (2004)

Cristiani, M., Cuel, R.: A Survey on Ontology Creation Methodologies. Idea Group Publishing, USA (2005)

Cruse, D.A.: Word Meanings and Concepts. In: Cruse, D.A. (ed.) Meaning in Language: an Introduction to Semantics and Pragmatics, pp. 125–140. Oxford University Press, Oxford (2004)

Dai, L.L., Huang, H.Y., And Chen, Z.X.: A Comparative Study on Feature Selection in Chinese Text Categorization. Journal of Chinese Information Processing 1, 26–32 (2004)

Daft, R.L.: Organization Theory and Design. Thomson/South-Western College Pub., U.S (2004)

Danesi, M.: Messages Signs and Meanings: a Basic Textbook in Semiotics and Communication Theory, 3rd edn. Canadian Scholars Inc., Toronto (2004)

Davenport, T.H., Prusak, L.: Working Knowledge. How Organizations Manage What They Know, 2nd edn. Harvard Business Press, U.S (2000)

Deng, J., Dong, W., Socher, R., Li, L.J., Fei, F.L.: Imagenet: a Large-scale Hierarchical Image Database. In: Proceedings of the IEEE Computer Sociaty Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, June 20-26, pp. 248–255 (2009)

Denny, M.: Ontology Editor Survey Result (2004), Retrieved from http://www.xml.com/2004/07/14/examples/Ontology_Editor_Survey_2004_Table_-_Michael_Denny.pdf

Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R.S., Peng, Y.: Swoogle: a Search and Metadata Engine for the Semantic Web. In: Proceedings of the 13th ACM International Conference on Information and Knowledge Management, Washington, DC, USA, pp. 652–659 (2004)

Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R.S., Peng, Y., Reddivari, P., Doshi, V.C., Sachs, J.: Swoogle: a Semantic Web Search and Metadata Engine. In: Proceedings of the 13th ACM Conference on Information and Knowledge Management (November 2004)

Ding, L., Pan, R., Finin, T.W., Joshi, A., Peng, Y., Kolari, P.: Finding and ranking knowledge on the semantic web. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 156–170. Springer, Heidelberg (2005)

Ding, Z., Peng, Y., Pan, R., Yu, Y.: A Bayesian Methodology Towards Automatic Ontology Mapping. In: Proceedings of the Workshop on Context & Ontologies. Twentieth Conference on Artificial Intelligence, AAAI (2005)

Dong, Z., Dong, Q.: Hownet (1999), Retrieved from http://www.keenage.com/TheoryandpracticeofHowNet/04.pdf

Dong, Z., Dong, Q.: Ontology and Hownet (2003), Retrieved from http://www.keenage.com/papers/Ontology&HowNet.ppt

Duch, W., Matykiewicz, P., Pestian, J.: Neurolinguistic Approach to Natural Language Processing With Applications to Medical Text Analysis. Neural Networks 21(10), 1500–1510 (2008)

Duda, R., Hart, P., Stork, D.: Pattern Classification, 2nd edn. John Wiley & Sons, Chichester (2001)

Duda, R.O., Hart, P.E.: Pattern Classification and Scene Analysis. Wiley, Chichester (1973)

E B.C.: Aristotle Iv. In: Barnes, J. (ed.) Complete Works of Aristotle. Princeton University Press, Princeton (1995)

Ehrig, M., Staab, S.: QOM – Quick Ontology Mapping. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 683–697. Springer, Heidelberg (2004)

Ehrig, M., Sure, Y.: Ontology Mapping - An Integrated Approach. In: Bussler, C.J., Davies, J., Fensel, D., Studer, R. (eds.) ESWS 2004. LNCS, vol. 3053, pp. 76–91. Springer, Heidelberg (2004)

El-Diraby, T.A., Lima, C., Feis, B.: Domain Taxonomy for Construction Concepts: Toward a Formal Ontology for Construction Knowledge. Journal of Computing in Civil Engineering 19(4), 394–406 (2005)

Endres, B.: Jatke: a Platform for the Integration of Ontology Learning Approaches (2005)

Ereshefsky, M.: The Poverty of the Linnaean Hierarchy: a Philosophical Study of Biological Taxonomy. Cambridge University Press, Cambridge (2000)

Fabrizio, S.X.: Machine Learning in Automated Text Categorization. ACM Computing Surveys (csur) 34(1), 1–47 (2002)

Farkas, C., Stoica, A.: Correlated Data Inference in Ontology Guided Xml Security Engine. In: IFIP 17th WG 11.3 Working Conference on Data and Application Security (2003)

Fellbaum, C.: WordNet: an Electronic Lexical Database. MIT Press, Cambridge (1998)

Foo, S., Li, H.: Chinese Word Segmentation and Its Effect on Information Retrieval. Information Processing and Management 40(1), 161–190 (2004)

Fox, C.: Information Retrieval: Data Structures and Algorithms. Prentice Hall, New Jersey (1992)

Fragos, K., Maistros, I., Skourlas, C.: Word Sense Disambiguation Using WordNet Relations. In: Proceedings of the 1st Balkan Conference in Informatics, Thessaloniki, Greece (2003)

Fu, S.X., Yuan, D.R., Huang, B.X., Zhong, Z.: Word Extraction Without Thesaurus Based on Statistics. Journal of Guangxi Academy of Sciences 18, 252–264 (2002)

Genesereth, M.R., Nilsson, N.J.: Logical Foundation of Artificial Intelligence. Margan Kaufmann, California (1987)

Gernsbacher, M.A.: Language Comprehension As Structure Building. Lawrence Erlbaum, Hillsdale (1990)

Gomez, P., Mariano, F.L., Fernhndez, L., Oscar, C.: Ontological Engineering: With Examples from the Areas of Knowledge Management E-commerce and the Semantic Web. Springer, Heidelberg (2004)

Gomez-Perez, A., Fernandez-Lopez, M., Corcho, O.: Ontological Engineering. With Examples from the Areas of Knowledge Management E-commerce and the Semantic Web. In: Advanced Information and Knowledge Processing, 1st edn. Springer, Heidelberg (2004)

GraphML: What Is Graphml? the Graphml File Format (2007), Retrieved from `http://graphml.graphdrawing.org/index.html`

Gruber, T.: A Translation Approach to Portable Ontologies. Knowledge Acquisition 5(2), 199–220 (1993)

Gruber, T.R.: A Translation Approach to Portable Ontologies. Knowledge Acquisition 5(2), 199–220 (1993)

Gruber, T.R.: A Translation Approach to Portable Ontology Specifications. Knowledge Acquisition 5(2), 199–220 (1993)

Gruber, T.R.: Toward Principles for the Design of Ontologies Used for Knowledge Sharing. International Journal of Human-computer Studies 43(5), 907–928 (1995)

Gruber, T.R.: Ontology. In: Liu, L., Ozsu, M.T. (eds.) Encyclopedia of Database Systems. Springer, Heidelberg 2008)

Gruninger, M., Fox, M.S.: Methodology for the Design and Evaluation of Ontologies. In: Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal (1995)

Guan, Y., Wang, X.L., Kong, X.Y.: Quantifying Semantic Similarity of Chinese Words from Hownet. In: Proceedings of the 1st International Conference on Machines Learning and Cybernetics, Beijing, China, November 4-5, pp. 234–239 (2002)

Guarino, N.: Formal Ontology Conceptual Analysis and Knowledge Representation. International Journal of Human-computer Studies 43, 625–640 (1995)

Guarino, N.: Formal Ontology and Information Systems. In: Proceedings of the 1st International Conference on Formal Ontologies in Information Systems, Trento, Italy (June 1998)

Guarino, N., Giaretta, P.: Ontologies and Knowledge Bases: Towards a Terminological Clarification. In: Mars, N. (ed.) Towards Very Large Knowledge Based: Knowledge Building and Knowledge Sharing, pp. 25–32. IOS Press, Amsterdam (1995)

Guarino, N., Giaretta, P.: Ontologies and Knowledge Bases: Towards a Terminological Clarification. In: Mars, N. (ed.) Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing, pp. 25–32. IOS Press, Amsterdam (1995)

Guarino, N., Masolo, C., Vetere, G.: Ontoseek: Content-based Access to the Web. IEEE Intelligent Systems 14(3), 70–80 (1999)

Guiraud, P.: Semiology. Routledge and Kegan Paul, London (1975)

Gulla, J.A., Borch, H.O., Ingvaldsen, J.E.: Ontology Learning for Search Applications. In: Proceedings of the 6th International Conference on Ontologies Databases and Applications of Semantics, ODBASE 2007 (2007)

Hou, T., Lan, G.Y.: Automatic Text Categorization For Chinese Web Pages. Computer and Communications 23(125), 114–116 (2005)

Haase, P., Volker, J.: Ontology Learning and Reasoning - Dealing With Uncertainty and Inconsistency. In: Proceedings of the International Semantic Web Conference Workshop 3: Uncertainty Reasoning for the Semantic Web (ISWC-URSW 2005), Galway, Ireland, pp. 45–55 (2005)

Haase, P., Völker, J.: Ontology Learning and Reasoning — Dealing with Uncertainty and Inconsistency. In: da Costa, P.C.G., d'Amato, C., Fanizzi, N., Laskey, K.B., Laskey, K.J., Lukasiewicz, T., Nickles, M., Pool, M. (eds.) URSW 2005 - 2007. LNCS (LNAI), vol. 5327, pp. 366–384. Springer, Heidelberg (2008)

Hahn, U., Schnattinger, K.: Towards Text Knowledge Engineering. In: Proceedings of the Fifteenth National/tenth Conference on Artificial Intelligence/innovative Applications of Artificial Intelligence, pp. 524–531 (1998)

Handschuh, S.: Annotation for the Semantic Web. IOS Press, Amsterdam (2003)

Handschuh, S., Staab, S.: Authoring and Annotation of Web Pages in Cream. In: Proceedings of the 11th International World Wide Web Conference (2002)

Handschuh, S., Staab, S.: Cream - Creating Metadata for the Semantic Web. Computer Networks 42, 579–598 (2004)

Harmelen, F.V., Lifschitz, V., Porter, B.: Handbook of Knowledge Representation. Elsevier, Amsterdam (2008)

Hartigan, J.A., Wong, M.A.: Ak-means Clustering Algorithm. Appl. Statist. 28, 100–108 (1979)

Hazman, M., Beltagy, S.R., Rafea, A.: Ontology Learning from Textual Web Documents. In: Proceedings of the 6th International Conference on Informatics and Systems (INFOS 2008), Giza, Egypt, pp. 113–120 (2008)

Hearst, M.A.: Text Data Mining: Issues. Techniques. and the Relationship to Information Access. Presented At the Uw/ms Workshop on Data Mining (1997)

Heflin, J., Hendler, J.: A Portrait of the Semantic Web. IEEE Intelligent Systems, 54–59 (2001)

Hendler, J.: Agents and the Semantic Web. IEEE Intelligent System, 18–25 (March/April 2001)

Hendler, J.: Agents and the Semantic Web. IEEE Intelligent Systems, 30–36 (2001)

Hendler, J.A.: Agents and the Semantic Web. IEEE Intelligent Systems 16(2), 30–37 (2001)

Hirst, G.: Ontology and the Lexicon. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies, pp. 209–230. Springer, Berlin (2004)

Hoogs, A., Rittscher, J., Stein, G., Schmiederer, J.: Video Content Annotation Using Visual Analysis and a Large Semantic Knowledgebase. In: IEEE Computer Soc. Conf. Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 327–334 (2003)

Hotho, A., Nürnberger, A., Paab, G.: A Brief Survey of Text Mining. Journal for Computational Linguistics and Language Technology 20(1), 19–62 (2005)

HowNet: Computation of Meaning (2003), Retrieved from http://www.keenage.com/

Hu, H., Du, X., Ouyang, J.H.: Ontology Learning Using WordNet Lexicon. In: Liu, G.R., Tan, V.B.C., Han, X. (eds.) Computational Methods, pp. 1249–1253. Springer, Netherlands (2006)

Hua, Z., Wang, X.J., Liu, Q., Lu, H.: Semantic Knowledge Extraction and Annotation for Web Images. In: Proceedings of the 13th Annual ACM International Conference on Multimedia, Singapore, pp. 467–470 (2005)

IEC SC 34, I: Home of Sc34/wg3 Information Association (2008), Retrieved from http://www.isotopicmaps.org/

IEEE SUO WG, Standard Upper Ontology Working Group (2003), Retrieved from http://suo.ieee.org/index.html

Ide, N., Vyronis, J.: Refining Taxonomies Extracted from Machine Readable Dictionaries. Research in Hummanities Computing 2, 145–159 (1994)

Jameson, F.: The Prison-house of Language. Princeton University Press, Princeton (1972)

Jenings, N.R., Wooldridge, M.: Applications of Intelligent Agents Agent Technology: Foundations Applications and Markets. Springer, Heidelberg (1998)

Ji, H., Tan, A.H.: Machine Learning Methods for Chinese Web Page Categorization. Annual Meeting of the ACL 1, 93–100 (2000)

Joachims, T.: A Probabilistic Analysis of the Rocchio Algorithm With Tfidf for Text Categorization. In: Proceedings of the 14th International Conference on Machine Learning, TN, USA, pp. 143–151 (1997)

Jones, K.S.: Notes and References on Early Automatic Classification Work. ACM Sigir Forum 25(1), 10–17 (1991)

Kalfoglou, Y., Schorlemmer, M.: If-map: an Ontology Mapping Method Based on Information Flow Theory. Journal on Data Semantics 1(1), 98–127 (2003)

Kalvans, J., Muresan, S.: Text Mining Techniques for Fully Automatic Glossary Construction. In: Proceedings of the Human Language Technology 2001 Conference, San Diego, CA (March 2001)

Karmakar, S.: Designing Domain Ontology: a Study in Lexical Semantics (2007)

Kitamura, Y., Kashiwase, M., Fuse, M.: Deployment of an Ontological Framework of Functional Design Knowledge. Artificial Intelligence for Engineering 18(2), 115–127 (2004)

Klusch, M.: Information Agent Technology for the Internet: a Survey. Journal on Data and Knowledge Engineering Special Issue on Intelligent Information Integration 36(3), 93–100 (2001)

Koenemann, J., Belkin, N.J.: A Case for Interaction: a Study of Interactive Information Retrieval Behavior and Effectiveness. In: Proceedings of the Sigchi Conference on Human Factors in Computing Systems: Common Ground, Vancouver, Canada, April 13-18, pp. 205–212 (1996)

Kohonen, T.: An Introduction to Neural Computing. Neural Networks 1(1), 3–16 (1988)

Kok, W.G., Ping, W.W.: Annotating Information Structures in Chinese Texts Using Hownet. In: Proceedings of the 2nd Workshop on Chinese Language Processing, Hong Kong, China, pp. 85–92 (October 2000)

Kong, J.: Ontology Learning for Chinese Information Organization and Knowledge Discovery in Ethnology and Anthropology. Data Science Journal 6(19), 500–510 (2007)

Kotis, K., Vouros, A.: Human-centered Ontology Engineering: the Hcome Methodology. Knowledge and Information Systems 10, 109–131 (2006)

Kotis, K., Vouros, G.A., Alonso, J.P.: Hcome: Tool-supported Methodology for Collaboratively Devising Living Ontologies. In: Bussler, C.J., Tannen, V., Fundulaki, I. (eds.) SWDB 2004. LNCS, vol. 3372, pp. 155–166. Springer, Heidelberg (2005)

Krieg, P.: What Makes a Thinking Machines? Computational Semiotics and Semiotic Computation. Semiotics and Intelligent Systems Development. Idea Group Publishing, USA (2007)

Kumar, A., Smith, B., Borgelt, C.: Dependence Relationships Between Gene Ontology Terms Based on Tigr Gene Product Annotations. In: Proceedings of the 3rd International Workshop on Computational Terminology (2004)

Kwon, O., Lee, J.: Text Categorization Based on K-nearest Neighbor Approach for Web Site Classification. Information Processing and Management 39, 25–44 (2003)

Li, P.: Godel Theorem and Semiotics. In: Proceedings of the Conference on Intelligent Systems and Semiotics, Gaithersburg, vol. 2, pp. 14–18 (1996)

Lakoff, G., Johnson, M.: Metaphors We Live by. University of Chicago Press, Illinois (1980)

Lam, S., Lee, L.: Feature Reduction for Neural Network Based Text Categorization. In: Proceedings 6th IEEE International Conference on Database Advanced System for Advanced Application, Hsinchu, Taiwan, April 19-21, pp. 195–202 (1999)

Lammari, N., Metais, E.: Building and Maintaining Ontologies: a Set of Algorithms. Data & Knowledge Engineering 48, 155–176 (2003)

Lancaster, F.W., Warner, A.J.: Information Retrieval Today. Information Resources Press, Arlington (1993)

Lawvere, F.W., Schanuel, S.H.: Conceptual Mathematics. Cambridge University Press, Cambridge (1997)

Lee, D.L., Chuang, H., Seamons, K.: Document Ranking and the Vector-space Model. IEEE Software, 67–75 (1997)

Lee, J.H.: A Fuzzy Ontology and Its Application to News Summarization. IEEE Transactions on Systems Man and Cybernetics 35(50), 859–888 (2005)

Lee, M.C., Tsai, K.H., Wang, T.I.: A Practical Ontology Query Expansion Algorithm for Semantic-aware Learning Objects Retrieval. Computers & Education 50, 1240–1257 (2008)

Lee, R.S.T.: Fuzzy-neuro Approach to Agent Applications: from the AI Perspective to Modern Ontology. Springer, Heidelberg (2005)

Lee, R.S.T., Liu, J.N.K.: Ijade Eminer - a Web-based Mining Agent Based on Intelligent Java Agent Development Environment (IJADE) on Internet Shopping. In: Proceedings of the 5th Pacific-asia Conference in Knowledge Discovery and Data Mining, Hong Kong, China, April 16-18, pp. 28–40 (2001)

Leenheer, P.D., Moor, A.D., Meersman, R.: Context Dependency Management in Ontology Engineering: a Formal Approach. Journal on Data Semantics 8, 26–56 (2007)

Lenat, D., Guha, R.: Building Large Knowledge-based Systems. Addison-Wesley, Reading (1990)

Li, F.: An Introduction to Semantics. Peking University Press (2006)

Li, Y., Zhong, N.: Capturing Evolving Patterns for Ontology-based Web Mining. In: Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence, Washington, DC, USA, Beijing China, pp. 256–263 (2004)

Li, Y., Chung, S.M., Holt, J.: Text Document Clustering Based on Frequent Word Meaning Sequences. Data and Knowledge Engineering 64(1), 381–404 (2008)

Li, Y., Lao, C., Chung, S.M.: Text Clustering With Feature Selection by Using Statistical Data. IEEE Transaction on Knowledge and Data Engineering 20(5), 641–652 (2008)

Lim, E.H.Y., Lee, R.S.T.: Ijade InfoSeeker: on Using Intelligent Context-aware Agents for Retrieving and Analyzing Chinese Web Articles. In: Lee, R.S.T., Loia, V. (eds.) Computational Intelligence for Agent-based Systems, pp. 127–153. Springer, Heidelberg (2007)

Lim, E.H.Y., Lee, R.S.T., Liu, J.N.K.: KnowledgeSeeker - an Ontological Agent-based System for Retrieving and Analyzing Chinese Web Articles. In: Proceedings of the IEEE International Conference on Fuzzy Systems, Hong Kong, China, June 1-6, pp. 1034–1041 (2008)

Lim, E.H.Y., Liu, J.N.K., Lee, R.S.T.: Knowledge Discovery from Text Learning for Ontology Modeling. In: Proceedings of the 6th International Conference on Fuzzy Systems and Knowledge Discovery, Tianjin, China, August 14-16, vol. 7, pp. 227–231 (2009)

Lim, E.H.Y., Tam, H.W.K., Wong, S.W.K., Liu, J.N.K., Lee, R.S.T.: Collaborative Content and User-based Web Ontology Learning System. In: Proceedings of the 18th International Conference on Fuzzy Systems, Jeju Island, Korea, August 20-24, pp. 1050–1055 (2009)

Lin, H.F., Ma, Z.Y., Yao, T.S.: Chinese Text Filtering Model Based on Semantic Frame. Journal of Computer Research and Development 38, 136–141 (2001)

Liu, K.: Semiotics in Information Systems Engineering. Cambridge University Press, Cambridge (2000)

Liu, T., Liu, S., Chen, Z., Ma, W.: An Evaluation on Feature Selection for Text Clustering. In: Proceedings of the International Conference on Machine Learning, Washington, DC, August 21-24, pp. 415–424 (2003)

Liu, T., Wu, Y., Wang, K.Z.: A Chinese Word Automatic Segmentation System Based on String Frequency Statistics Combined With Word Matching. Journal of Chinese Information Processing 12, 17–25 (1998)

Liu, Y., Loh, H.T., Sun, A.: Imbalanced Text Classification: a Term Weighting Approach. Expert Systems With Applications (ESWA) 36(1), 690–701 (2009)

Louwerse, M.: An Analytic and Cognitive Parameterization of Coherence Relations. Cognitive Linguistics, 291–315 (2002)

Louwerse, M.M.: An Analytic and Cognitive Parameterization of Coherence Relations. Cognitive Linguistics 12, 291–315 (2002)

Maarek, Y.S., Smadja, F.Z.: Full Text Indexing Based on Lexical Relations an Application: Software Libraries. In: Proceedings of the 12th Annual International ACM Sigir Conference on Research and Development in Information Retrieval, Massachusetts, United States, June 25-28, pp. 198–206 (1989)

Maedche, A.: Ontology Learning for the Semantic Web. IEEE Intelligent Systems 16(2), 72–79 (2001)

Maedche, A.: Ontology Learning for the Semantic Web. Kluwer Academic Publishers, Norwell (2002)

Maedche, A., Staab, S.: Ontology Learning for the Semantic Web. IEEE Intelligent Systems, 72–79 (2001)

Maedche, A., Staab, S.: Measuring Similarity between Ontologies. In: Gómez-Pérez, A., Benjamins, V.R. (eds.) EKAW 2002. LNCS (LNAI), vol. 2473, pp. 251–263. Springer, Heidelberg (2002)

Magkanaraki, A., Karvounarakis, G., Anh, T.T., Christophides, V., Plexousakis, D.: Ontology Storage and Querying. ICS-forth. Technical Report 308 (2002)

Mahinovs, A., Tiwari, A.: Text Classification Method Review. In: Aoy, R., Baxter, D. (eds.) Decision Engineering Report Series, pp. 1–13. Cranfield University, United Kingdom (2007)

Mancini, C., Shum, S.J.B.: Modelling Discourse in Contested Domains: a Semiotic and Cognitive Framework. International Journal of Human-computer Studies 64(11), 1154–1171 (2006)

Manning, C., Schutze, H.: Foundations of Statistical Natural Language Processing. MIT Press, Cambridge (1999)

Martin, P., Eklund, P.W.: Knowledge Retrieval and the World Wide Web. IEEE Intelligent System, 18–25 (May/June 2000)

Meadow, C.T.: Text Information Retrieval Systems. Academic Press Inc., Orlando (1992)

Mesleh, A.M.: Chi Square Feature Extraction Based Svms Arabic Language Text Categorization System. In: Proceedings of the 2nd International Conference on Software and Data Technologies, Barceolona, Spain, July 22-25, pp. 235–240 (2007)

Miller, A.: WordNet: an On-line Lexical Resource. Journal Lexicography 3(4), 1–1 (1990)

Miller, G.A.: WordNet: a Lexical Database for English. Communications of the ACM 38(11), 39–41 (1995)

Miller, G.A.: WordNet: an Electronic Lexical Database. MIT Press, U.S (1998)

Minick, N.: The Development of Vygotsky Thought: an Introduction. In: Vygotsky, L.S. (ed.) The Collected Works of L.s. Vygotsky. Problem of General Psychology, vol. 1, pp. 17–36. Plenum Press, New York (1987)

Mitchell, T.M.: Machine Learning. McGraw-Hill Higher Education, New York (1997)

Mitra, P., Wiederhold, G., Jannink, J.: Semi-automatic Integration of Knowledge Sources. In: Proceedings of the International Conference of Information Fusion (1999)

Muller, J.: Hierarchical Models in Semiotics and Psychoanalysis. In: Muller, J., Brent, J. (eds.) Peirce. Semiotics and Psychoanalysis. The Johns Hopkins University Press, Baltimore Maryland (2000)

Navigli, R., Velardi, P.: Learning Domain Ontologies from Document Warehouses and Dedicated Web Sites. Computational Linguistics 30(2), 151–179 (2004)

Navigli, R., Velardi, P.: From Glossaries to Ontologies: Extracting Semantic Structure from Textual Definitions. Frontiers in Artificial Intelligence and Applications 167, 71–89 (2008)

Neches, R., Fikes, R., Finin, T., Gruber, T., Patil, R., Senatir, T.: Enabling Technology for Knowledge Sharing. AI Magazine 12(3), 36–56 (1991)

Neuman, Y.: A Theory of Meaning. Information Sciences 176, 1435–1449 (2006)

Newell, A.: The Knowledge Level. The AI Magazine, 1–20 (Summer 1981)

Ni, X., Xue, G.R., Ling, X., Yu, Y., Yang, Q.: Exploring in the Weblog Space by Detecting Informative and Affective Articles. In: Proceedings of the 16th International World Wide Web Conference, Banff, Alberta, Canada, May 8-12, pp. 281–290 (2007)

Nicole, A.D., Missikoff, M., Navigli, R.: A Software Engineering Approach to Ontology Building. Information Systems 34, 258–275 (2009)

Niles, I., Pease, A.: Towards a Standard Upper Ontology. In: Proceedings of the International Conference on Formal Ontology in Information Systems, Ogunquit, Maine, USA, October 17-19, pp. 2–9 (2001)

Nilsson, N.: Logic and Artificial Intelligent. Journal of Artificial Intelligence Research, 31–35 (1991)

Nosek, J.T., Roth, I.: A Comparison of Formal Knowledge Representation Schemes As Communication Tools: Predicate Logic Versus Semantic Network. International Journal on Man-machine Studies 33, 227–239 (1990)

Nyamsuren, E., Choi, H.J.: Building a Semantic Model of a Textual Document for Efficient Search and Retrieval. In: Proceedings of the 11th International Conference on Advanced Communication Technology, Gangwon-Do, South Korea, February 15-18, pp. 298–302 (2009)

OKBC, Open Knowledge Base Connectivity Home Page (1995), Retrieved from `http://www.ai.sri.com/~okbc/`

OWL, Owl Web Ontology Language Guide (2004), Retrieved from `http://www.w3.org/TR/owl-guide/`

Obrst, L.: Ontologies for Semantically Interoperable Systems. In: Proceedings of the 12th International Conference on Information and Knowledge Management, New Orleans, Louisiana, USA, November 2-8, pp. 366–369 (2003)

Oddy, R.N.: Information Retrieval Research. Butterworths, London (1981)

OpenCyc, Opencyc (2003), Retrieved from `http://www.opencyc.org/`

Pan, R., Ding, Z., Yu, Y., Peng, Y.: A Bayesian Network Approach to Ontology Mapping. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 563–577. Springer, Heidelberg (2005)

Park, Y., Byrd, R.J., Boguraev, B.K.: Automatic Glossary Extraction: Beyond Terminology Identification. In: Proceedings of the 19th International Conference on Computational Linguistics, Taipei, Taiwan, August 24-September 1, pp. 772–778 (2002)

Patel, C., Supekar, K., Lee, Y.Y., Park, E.K.: Ontokhoj: a Semantic Web Portal for Ontology Searching Ranking and Classification. In: Proceedings of the 5th International Workshop on Web Information and Data Management, New Orleans, Louisiana, USA, November 07-08, pp. 58–61 (2003)

Patel, M., Duke, M.: Knowledge Discovery in an Agents Environment. In: Proceedings of the European Semantic Web Symposium 2004, Heraklion, Greece, May 10-12, pp. 121–136 (2004)

Patwardhan, S., Pedersen, T.: Using WordNet-based Context Vectors to Estimate the Semantic Relatedness of Concepts. In: Proceedings of the Conference European Chapter of the Association for Computational Linguistics, EACL (2006)

Perlovsky, L.: The Knowledge Instinct. Basic Books, New York (2006)

Philippe, M.: Using the WordNet Concept Catalog and a Relation Hierarchy for Knowledge Acquisition. Proceedings of the National Academy of Sciences 99(3), 1742–1747 (1995)

Pinto, H.S., Staab, S., Tempich, C.: Diligent: Towards a Fine-grained Methodology for Distributed Loosely-controlled and Evolving Engineering of Ontologies. In: Proceedings of the 16th European Conference on Artificial Intelligence (ECAI 2004), Valencia (August 2004)

Pirr, G., Talia, D.: Ufome: a User Friendly Ontology Mapping Environment. In: Proceedings of the Fourth Italian Swap Workshop on Semantic Web Applications and Perspectives (2007)

Polpinij, J., Ghose, A.K.: An Ontology-based Sentiment Classification Methodology for Online Consumer Reviews. Web Intelligence 2008, 518–524 (2008)

Ponte, J., Croft, W.: A Language Modeling Approach to Information Retrieval. In: Proceedings of the 21st Annual International ACM Sigir Conference on Research and Development in Information Retrieval, New York, USA, pp. 275–281 (1998)

Pretorius, A.J.: Ontologies - Introduction and Overview. In: Pretorius, A.J. (ed.) Lexon Visualisation: Visualising Binary Fact Types in Ontology Bases, pp. 1–13. Vrije Universiteit Brussel (2004)

Protégé: The Protégé Ontology Editor and Knowledge Acquisition System (2009), Retrieved from `http://protege.stanford.edu/`

Quinlan, J.R.: Induction of Decision Trees. Machine Learning 1, 81–106 (1986)

RDF, W3c Resource Description Framework (2007), Retrieved from `http://www.w3.org/RDF/`

Randall, D., Schrobe, H., Szolovits, P.: What Is a Knowledge Representation? AI Magazine 14(1), 17–33 (1993)

Reynaud, C., Safar, B.: Exploiting WordNet As Background Knowledge. In: Shvaiko, P., Euzenat, J., Giunchiglia, F., He, B. (eds.) Proceedings of the International Workshop Ontology Matching, OM 2007 (2007)

Rezgui, Y.: Text Based Domain Ontology Building Using Tf-idf and Metric Clusters Techniques. Knowledge Engineering Review 22(4), 379–403 (2007)

Rifaieh, R., Benharket, A.N.: From Ontology Phobia to Contextual Ontology Use in Enterprise Information Systems. In: Taniar, D., Rahayu, J.W. (eds.) Web Semantics & Ontology, pp. 115–165. Idea Group Publishing, Hershey (2006)

Rijsbergen, C.J.V.: Information Retrieval, 2nd edn. Butterworth, London (1979)

Roche, C.: Lexical and Conceptual Structures in Ontology. In: Ali, M., Dapoigny, R. (eds.) Advances in Applied Artificial Intelligence, pp. 1034–1041. Springer, Heidelberg (2006)

Rogati, M., Yang, Y.: High-performing Feature Selection for Text Classification. In: Proceedings of the 11th International Conference on Information and Knowledge Management, McLean, Virginia, USA, pp. 659–661 (2002)

Russell, B.: Introduction to Mathematical Philosophy. George Allen and Unwin, London (1919)

SUMO Ontology, Suggested Upper Merged Ontology, sumo (2004), Retrieved from `http://www.ontologyportal.org/`

SUO 4D Ontology, Develop an Ontology Based on the 4-dimensional Paradigm (2005), Retrieved from `http://www.tc184-sc4.org/wg3ndocs/wg3n1328/lifecycle_integration_schema.html`

Salto, G., McGill, M.J.: An Introduction to Modern Information Retrieval. McGraw-Hill, New York (1983)

Salton, G.: Automatic Information Organization and Retrieval. McGraw-Hill, New York (1968)

Salton, G.: Another Look At Automatic Text-retrieval Systems. Communications of the ACM 29(7), 648–656 (1986)

Salton, G.: Automatic Text Processing: the Transformation Analysis and Retrieval of Information by Computer. Addison-Wesley Longman Publishing Co. Inc., Boston (1989)

Salton, G.: The State of Retrieval System Evaluation. Information Processing and Management: an International Journal 28(4), 441–449 (1992)

Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. McGraw-Hill Inc., New York (1986)

Salton, G., Wong, A., Yang, C.S.: A Vector Space Model for Automatic Indexing. Communications of the ACM 18(11), 613–620 (1975)

Salton, G., Wong, A., Yang, C.S.: A Vector Space Model for Automatic Indexing. Communications of the ACM 18(11), 613–620 (1998)

Sanchez, D., Moreno, A.: Creating Ontologies from Web Documents. In: Vitri, J., Radeva, P., Aguil, I. (eds.) Recent Advances in Artificial Intelligence Research and Development (Proceedings of Set Congr Artificial (CCIA 2004), pp. 11–18. IOS Press, Barcelona Catalunya (2004)

Saussure, F.D.: Course in General Linguistics (r. Harris. Trans.). Duckworth, London (1972)

Savage, L.J.: The Foundations of Statistics. Wiley, New York (1954)

Scholtes, J.C.: Neural Networks in Natural Language Processing and Information Retrieval. North-Holland, The Netherlands (1993)

Schreiber, A.T., Dubbeldam, B., Wielemaker, J., Wielinga, B.: Ontology-based Photo Annotation. IEEE Intelligent Systems 16(3), 66–74 (2001)

Schreiber, Z.: Semantic Information Management. White Paper, Unicorn (2003)

Schwarz, U., Smith, B.: Ontological Relations. In: Reicher, M.E., Seibt, J., Smith, B., Wachter, D.V. (eds.) Applied Ontology: an Introduction. Ontos Verlag, Heusenstamm (2008)

Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computing Surveys 34(1), 1–47 (2002)

Sedding, J., Kazakov, D.: WordNet-based Text Document Clustering. In: Proceedings of the 3rd Workshop on Robust Methods in Analysis of Natural Language Data (romand), Geneva, Switzerland (2004)

Sidhu, A.S., Dillon, T.S.: Knowledge Discovery in Biomedical Data Facilitated by Domain Ontologies. In: Zhu, X., Davidson, I. (eds.) Knowledge Discovery and Data Mining: Challenges and Realities With Real World Data. Idea Group, USA (2006)

Smith, B.: New Desiderata for Biomedical Terminologies. In: Reicher, M.E., Seibt, J., Smith, B., Wachter, D.V. (eds.) Applied Ontology: an Introduction. Ontos Verlag, Heusenstamm (2008)

Song, M., Wu, Y.F.: Handbook of Research on Text and Web Mining Technologies. Idea Group Inc, IGI Global (2009)

Soo, V.W., Lee, C.Y., Li, C.C., Chen, S.L., Chen, C.C.: Automated Semantic Annotation and Retrieval Based on Sharable Ontology and Case-based Learning Techniques. In: Proceedings of Joint Conference on Digital Libraries 2003, Rice University, Houston, Texas, pp. 61–72 (2003)

Soo, V.W., Lee, C.Y., Li, C.C., Chen, S.L., Chen, C.C.: Automated Semantic Annotation and Retrieval Based on Sharable Ontology and Case-based Learning Techniques. In: Proceedings of the 3rd ACM/IEEE-CS Joint Conference on Digital Libraries, Houston, Texas, May 27-31, pp. 61–72 (2003)

Sowa, J.F.: Conceptual Graphs. IBM Journal of Research and Development 20(4), 336–357 (1976)

Sowa, J.F.: Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley, Reading (1984)

Sowa, J.F.: Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley, Boston (1984)

Sowa, J.F.: Knowledge Representation. Logical Philosophical and Computational Foundations. Brooks/Cole, California (2000)

Staab, S., Studer, R.: Handbook on Ontologies. Springer, Heidelberg (2004)

Stumme, G., Maedche, A.: Fca-merge: Bottom-up Merging of Ontologies. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), pp. 225–234 (2009)

Su, X., Gulla, J.A.: An Information Retrieval Approach to Ontology Mapping. Data & Knowledge Engineering 58, 47–69 (2006)

Tague, J., Schultz, R.: Evaluation of the User Interface in an Information Retrieval System: a Model. Information Processing and Management: an International Journal 25(4), 377–389 (1989)

Tang, J., Li, J., Liang, B., Huang, X., Li, Y., Wang, K.: Using Bayesian Decision for Ontology Mapping. Web Semantics: Science/ Services and Agents on the World Wide Web 4(4), 243–262 (2006)

Tho, Q.T., Hui, S.C., Fong, A.C.M., Cao, T.H.: Automatic Fuzzy Ontology Generation for Semantic Web. IEEE Transactions on Knowledge and Data Engineering 18(6), 842–856 (2006)

Tian, X., Du, X., Hu, H., Li, H.: Modeling Individual Cognitive Structure in Contextual Information Retrieval. Computers and Mathematics With Applications 57, 1048–1056 (2009)

Tvarožek, M., Barla, M., Bieliková, M.: Personalized Presentation in Web-Based Information Systems. In: van Leeuwen, J., Italiano, G.F., van der Hoek, W., Meinel, C., Sack, H., Plášil, F. (eds.) SOFSEM 2007. LNCS, vol. 4362, pp. 796–807. Springer, Heidelberg (2007)

Ueberall, M., Drobnik, O.: On Topic Map Templates and Traceability. In: Proceedings of the 2nd International Conference on Topic Maps Research and Applications, Leipzig, Germany, October 11-12 (2006)

Uschold, M., King, M.: Towards a Methodology for Building Ontologies. In: Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing in Conjunction With IJCAI 1995, Montreal, Canada (1995)

Ushold, M., Gruninger, M.: Ontologies: Principles Methods and Applications. The Knowledge Engineering Review 11(2), 93–155 (1996)

Valerio, A., Leake, D.: Jump-starting Concept Map Construction With Knowledge Extracted from Documents. In: Proceedings of the Second International Conference on Concept Mapping, pp. 296–303 (2006)

Vallet, D., Cantador, I., Fernandez, M., Castells, P.: A Multi-purpose Ontology-based Approach for Personalized Content Filtering and Retrieval. In: Proceedings of the 1st Semantic Media Adaptation and Personalization 2006, Athens, Greece, December 26, pp. 19–24 (2006)

Velardi, P., Cucchiarelli, A., Petit, M.: A Taxonomy Learning Method and Its Application to Characterize a Scientific Web Community. IEEE Transactions on Knowledge and Data Engineering 19(2), 180–191 (2007)

Vogt, P.: Anchoring of Semiotic Symbols. Robotics and Autonomous Systems 43, 109–120 (2003)

W3C Semantic Web (2007), Retrieved from `http://www.w3.org/2001/sw/`

W3C Semantic Web, W3c Semantic Web Activity (2009), Retrieved from `http://www.w3.org/2001/sw/`

Wang, W., Nie, J.: A Latent Semantic Structure Model for Text Classification. In: Proceedings of the ACM Sigir 2003 Workshop on Mathematical/formal Methods in Information Retrieval, Toronto, Canada, August 1 (2003)

Widyantoro, D.H., Yen, J.: Relevant Data Expansion for Learning Concept Drift from Sparsely Labeled Data. IEEE Transactions on Knowledge and Data Engineering 17(3), 401–412 (2005)

Wilcock, G.: Talking Owls: Towards an Ontology Verbalizer. In: Proceedings of the Conference on Human Language Technology for the Semantic Web and Web Services ISWC 2003, Sanibel Island, Florida, pp. 109–112 (2003)

Wisetphanichkij, S., Dejhan, K., Cheevasuvit, F., Mitatha, S., Arungsrisangchai, I., Yimman, S.: A Fusion Approach of Multi-spectral With Sar Image for Flood Area Analysis. In: Proceedings of the Asian Conference on Remote Sensing, Hong Kong, China, November 22-25, pp. 53–58 (1999)

Wong, S.W.K., Tam, H.W.K., Lim, E.H.Y., Liu, J.N.K., Lee, R.S.T.: The Multi-audiences Intelligent Online Presentation System. In: Proceedings of the 18th International Conference on Fuzzy Systems, Jeju Island, Korea, August 20-24, pp. 1863–1868 (2009)

Wooldridge, M., Jennings, N.: Intelligent Agents: Theory and Practice. The Knowledge Engineering Review 10(2), 115–152 (1995)

Yan, J.: Conceptual Modeling of Collaborative Manufacturing for Customized Products: an Ontological Approach. In: Proceedings of the Pacific Asia Conference on Information Systems, Suzhou, China (2008)

Yang, D.: Product Configuration Knowledge Modeling Using Ontology Web Language. Expert Systems With Applications 36(3), 4399–4411 (2009)

Yang, S.Y.: An Ontological Website Models-supported Search Agent for Web Services. Expert Systems With Applications 35(4), 2056–2073 (2008)

Yang, Y., Pedersen, J.O.: A Comparative Study on Feature Selection in Text Categorization. In: Proceedings of the 14th International Conference on Machine Learning, San Francisco, CA, USA, July 8-12, pp. 412–420 (1997)

yED. Yed Graph Editor (2009), Retrieved from
`http://www.yworks.com/en/products_yed_about.html`

yFile XML Extension: Graphml Extension Package (2009), Retrieved from
`http://www.yworks.com/en/products_yfiles_ep_graphml.html`

Yu, W., Liu, Y.: Automatic Identification of Semantic Relationships for Manufacturing Information Management. In: Cheng, K., Makatsoris, H., Harrison, D. (eds.) Proceedings of the 6th International Conference on Manufacturing Research (ICMR 2008). Brunel University, UK (2008)

Yuan, X.Y., Wang, T., Zhou, H.P., Xiao, J.: Constructing Ontology-based Requirement Model in Chinese Information Filtering. Journal of Chinese Information Processing 20(3), 63–64 (2006)

Zhang, N., Jia, Z.Y., Shi, Z.Z.: Text Categorization With KNN Algorithm. Computer Engineering 31(8), 127–130 (2005)

Zhang, C., Hao, T.: The State of the Art and Difficulties in Automatic Chinese Word Segmentation. Journal of System Simulation 1, 138–147 (2005)

Zhang, X., Li, H., Qu, Y.: Finding important vocabulary within ontology. In: Mizoguchi, R., Shi, Z.-Z., Giunchiglia, F. (eds.) ASWC 2006. LNCS, vol. 4185, pp. 106–112. Springer, Heidelberg (2006)

Zhao, Y., Halang, W., Wang, X.: Rough Ontology Mapping in E-business Integration. In: Kacprzyk, J. (ed.) E-service Intelligence Methodologies Technologies and Applications, pp. 75–93. Springer, Heidelberg (2007)

Zheng, Z., Srihari, R., Srihari, S.: A Feature Selection Framework for Text Filtering. In: Proceedings of the 3rd International Conference on Data Mining, Melbourne, Florida, USA, November 19-22, p. 705 (2003)

Appendix

Table A.1 Top 30 ranked terms selected in Domain Ontology Graph (文藝)

Rank	Term	Class	Count	x2	R	POS
1	創作	文藝	100	1052.387	11.003	VERB,NOUN
2	藝術	文藝	128	994.220	8.337	ADJ,NOUN
3	演出	文藝	103	989.608	10.151	VERB
4	作品	文藝	98	953.037	10.284	NOUN
5	觀眾	文藝	83	615.711	8.251	NOUN
6	藝術家	文藝	57	571.245	10.767	NOUN
7	文化	文藝	124	535.291	5.205	ADJ,NOUN
8	演員	文藝	52	495.439	10.339	NOUN
9	劇團	文藝	47	485.407	11.097	NOUN
10	節目	文藝	60	475.555	8.831	NOUN
11	音樂	文藝	68	470.187	7.864	NOUN
12	歌舞	文藝	48	452.354	10.264	VERB
13	劇院	文藝	47	431.713	10.050	NOUN
14	晚會	文藝	45	420.203	10.200	NOUN
15	戲劇	文藝	42	419.831	10.818	NOUN
16	文化部	文藝	42	419.831	10.818	NOUN
17	舞蹈	文藝	47	395.574	9.345	NOUN
18	文藝	文藝	93	390.077	5.244	NOUN
19	舉辦	文藝	80	389.494	5.926	VERB
20	表演	文藝	66	370.552	6.679	VERB
21	舞台	文藝	51	369.901	8.257	NOUN
22	電影	文藝	50	347.380	7.981	NOUN
23	歌曲	文藝	35	344.813	10.721	NOUN
24	劇目	文藝	32	336.121	11.333	NOUN
25	戲曲	文藝	32	336.121	11.333	NOUN
26	精品	文藝	36	334.422	10.200	NOUN
27	美術	文藝	49	331.677	7.822	NOUN
28	風格	文藝	39	320.784	9.208	NOUN
29	演唱	文藝	33	312.818	10.389	VERB
30	展覽	文藝	45	310.869	7.969	VERB,NOUN

Table A.2 Top 30 ranked terms selected in Domain Ontology Graph (政治)

Rank	Term	Class	Count	x2	R	POS
1	總統	政治	141	459.191	4.178	NOUN
2	訪問	政治	145	395.355	3.722	VERB
3	主席	政治	143	319.257	3.305	NOUN
4	外交	政治	90	313.974	4.476	ADJ
5	會見	政治	89	300.943	4.387	VERB
6	友好	政治	100	299.371	4.037	ADJ,NOUN
7	外長	政治	71	294.431	5.071	NOUN
8	總理	政治	97	282.511	3.973	NOUN
9	會談	政治	80	279.602	4.502	VERB,NOUN
10	外交部	政治	57	243.558	5.205	NOUN
11	和平	政治	95	214.523	3.414	ADJ
12	關系	政治	138	190.323	2.571	NOUN,VERB
13	議會	政治	47	190.247	5.035	NOUN
14	領導人	政治	77	187.957	3.605	NOUN
15	今天	政治	219	171.692	1.912	NOUN
16	部長	政治	103	168.426	2.855	NOUN
17	雙邊	政治	41	162.011	4.965	ADJ
18	雙方	政治	90	161.852	3.020	NOUN
19	表示	政治	141	155.348	2.317	VERB,NOUN
20	阿拉伯	政治	47	145.473	4.223	ADJ,NOUN
21	會晤	政治	35	129.196	4.755	VERB
22	邊關	政治	29	128.153	5.385	NOUN
23	抵達	政治	43	128.083	4.130	VERB
24	大使	政治	44	125.454	4.018	NOUN
25	委員長	政治	31	120.652	4.934	NOUN
26	舉行	政治	143	117.203	2.053	VERB
27	巴勒斯坦	政治	30	111.087	4.775	ADJ,NOUN
28	共和國	政治	65	106.731	2.920	NOUN
29	外交部長	政治	23	106.365	5.571	NOUN
30	和平共處	政治	24	104.724	5.348	VERB

Table A.3 Top 30 ranked terms selected in Domain Ontology Graph (交通)

Rank	Term	Class	Count	x2	R	POS
1	運輸	交通	71	537.419	8.486	VERB
2	鐵路	交通	61	527.658	9.547	NOUN
3	公路	交通	52	379.165	8.337	NOUN
4	車輛	交通	48	377.266	8.888	NOUN
5	交通	交通	89	366.682	5.177	NOUN
6	公交	交通	29	318.992	11.914	NOUN
7	旅客	交通	27	318.849	12.677	NOUN
8	列車	交通	28	306.618	11.874	NOUN
9	不忍	交通	24	295.112	13.147	VERB
10	客運	交通	26	281.935	11.787	VERB
11	仁慈	交通	24	281.511	12.621	ADJ
12	堅韌不拔	交通	24	281.511	12.621	ADJ
13	客車	交通	25	269.629	11.738	NOUN
14	交通部	交通	23	269.095	12.599	NOUN
15	行駛	交通	25	258.735	11.333	VERB
16	運量	交通	21	257.826	13.147	NOUN
17	公安	交通	37	237.243	7.600	NOUN
18	貨運	交通	23	234.428	11.199	VERB
19	鐵道	交通	23	234.428	11.199	NOUN
20	公安部	交通	28	233.253	9.439	NOUN
21	星期二	交通	24	227.171	10.517	NOUN
22	駕駛員	交通	22	222.328	11.124	NOUN
23	通車	交通	20	219.678	11.952	VERB
24	駕駛	交通	26	185.557	8.337	VERB
25	鐵道部	交通	17	182.958	11.763	NOUN
26	公安廳	交通	16	182.611	12.373	NOUN
27	路局	交通	14	171.269	13.147	NOUN
28	違章	交通	15	158.622	11.600	VERB
29	通行	交通	20	157.675	9.067	ADJ
30	車站	交通	16	150.671	10.517	NOUN

Table A.4 Top 30 ranked terms selected in Domain Ontology Graph (教育)

Rank	Term	Class	Count	x2	R	POS
1	教師	教育	103	1022.033	10.468	NOUN,ADJ
2	學校	教育	124	1000.659	8.630	NOUN
3	教學	教育	95	978.203	10.862	VERB
4	學生	教育	117	808.840	7.605	NOUN
5	辦學	教育	60	690.954	12.195	VERB
6	中學	教育	65	573.673	9.678	NOUN
7	培養	教育	86	566.989	7.491	VERB
8	教育	教育	132	491.253	4.568	NOUN,VERB
9	素質	教育	80	487.658	7.065	NOUN
10	小學	教育	63	474.540	8.492	NOUN
11	校長	教育	43	468.292	11.715	NOUN
12	師資	教育	38	457.412	12.805	NOUN
13	校園	教育	49	442.533	9.960	NOUN
14	高中	教育	43	415.982	10.589	NOUN
15	課程	教育	38	396.288	11.316	NOUN
16	畢業	教育	55	377.352	7.913	VERB
17	教材	教育	35	360.142	11.205	NOUN
18	家教	教育	34	358.921	11.457	NOUN
19	家長	教育	40	354.354	9.850	NOUN
20	學習	教育	75	351.621	5.785	VERB
21	課堂	教育	31	345.821	12.029	NOUN
22	德育	教育	27	323.165	12.805	NOUN
23	大學	教育	77	321.825	5.301	NOUN
24	初中	教育	34	310.055	10.125	NOUN
25	老師	教育	35	304.975	9.743	NOUN
26	高等	教育	42	301.271	8.274	ADJ
27	教委	教育	30	300.383	10.976	NOUN
28	教職工	教育	25	298.919	12.805	NOUN
29	師生	教育	33	298.469	10.061	NOUN
30	學科	教育	40	297.763	8.537	NOUN

Table A.5 Top 30 ranked terms selected in Domain Ontology Graph (環境)

Rank	Term	Class	Count	x2	R	POS
1	污染	環境	74	669.921	9.857	VERB
2	生態	環境	55	522.582	10.395	NOUN
3	環保	環境	48	485.245	11.005	VERB
4	保護	環境	75	296.599	5.092	ADJ,VERB
5	森林	環境	27	272.116	11.106	NOUN
6	排放	環境	26	269.028	11.363	VERB
7	污染物	環境	21	248.245	12.770	NOUN
8	廢水	環境	18	221.688	13.250	NOUN
9	大氣	環境	26	212.286	9.324	NOUN
10	環境	環境	95	207.438	3.313	NOUN
11	環保局	環境	16	195.357	13.163	NOUN
12	污染源	環境	14	183.101	13.986	NOUN
13	自然	環境	47	167.067	4.869	ADJ,NOUN,ADV
14	野生	環境	12	156.784	13.986	ADJ
15	污水	環境	13	156.013	12.987	NOUN
16	資源	環境	52	147.280	4.156	NOUN
17	垃圾	環境	18	146.061	9.324	NOUN
18	水源	環境	13	143.956	12.121	NOUN
19	動物	環境	24	143.813	7.297	NOUN
20	野生動物	環境	11	143.645	13.986	NOUN
21	水污染	環境	11	143.645	13.986	VERB,NOUN
22	流域	環境	14	136.493	10.878	NOUN
23	人類	環境	33	134.230	5.430	NOUN
24	地球	環境	20	134.142	7.992	NOUN
25	水質	環境	10	130.520	13.986	NOUN
26	土壤	環境	15	118.193	9.121	NOUN
27	綠色	環境	21	115.068	6.830	ADJ
28	回收	環境	14	113.253	9.324	VERB
29	防治	環境	21	111.615	6.675	VERB
30	植物	環境	15	101.380	8.069	NOUN

Table A.6 Top 30 ranked terms selected in Domain Ontology Graph (經濟)

Rank	Term	Class	Count	x2	R	POS
1	增長	經濟	116	486.090	5.145	VERB
2	出口	經濟	77	435.482	6.594	NOUN,VERB
3	企業	經濟	135	413.430	4.068	NOUN
4	收入	經濟	74	358.016	5.872	VERB,NOUN
5	市場	經濟	127	354.801	3.854	NOUN
6	銀行	經濟	67	334.633	6.036	NOUN
7	財政	經濟	70	333.673	5.822	NOUN
8	美元	經濟	84	304.704	4.749	NOUN
9	金融	經濟	53	297.719	6.644	NOUN
10	消費	經濟	67	296.086	5.519	VERB
11	產品	經濟	105	290.035	3.898	NOUN
12	投資	經濟	96	286.826	4.131	VERB
13	下降	經濟	76	286.683	4.905	VERB
14	百分之	經濟	86	277.108	4.375	ADJ
15	商品	經濟	63	276.431	5.504	NOUN
16	生產	經濟	121	275.848	3.409	VERB
17	同期	經濟	46	274.412	6.980	NOUN
18	增長率	經濟	36	262.151	8.194	NOUN
19	資本	經濟	46	256.524	6.631	NOUN
20	經濟學	經濟	41	255.544	7.237	NOUN
21	貿易	經濟	65	244.498	4.932	NOUN
22	價格	經濟	64	244.421	4.987	NOUN
23	季度	經濟	41	242.572	6.953	ADJ
24	幅度	經濟	55	241.390	5.531	NOUN
25	貨幣	經濟	41	236.466	6.820	NOUN
26	增長速度	經濟	34	230.024	7.739	NOUN
27	總額	經濟	38	226.067	6.993	NOUN
28	宏觀經濟	經濟	29	225.135	8.649	NOUN
29	通貨	經濟	33	222.223	7.714	NOUN
30	大幅	經濟	47	210.875	5.646	ADJ

Table A.7 Top 30 ranked terms selected in Domain Ontology Graph (軍事)

Rank	Term	Class	Count	x2	R	POS
1	武器	軍事	73	545.951	8.357	NOUN
2	作戰	軍事	54	469.320	9.563	ADJ,VERB
3	戰斗	軍事	58	454.864	8.764	ADJ,VERB
4	美軍	軍事	47	431.713	10.050	NOUN
5	導彈	軍事	47	431.713	10.050	NOUN
6	海軍	軍事	46	430.904	10.222	NOUN
7	部隊	軍事	67	427.040	7.372	NOUN
8	飛行	軍事	42	398.281	10.348	VERB
9	艦船	軍事	37	389.644	11.333	NOUN
10	國防	軍事	56	376.112	7.740	NOUN
11	發射	軍事	37	345.024	10.228	VERB
12	坦克	軍事	35	334.048	10.439	NOUN
13	空軍	軍事	40	331.108	9.252	NOUN
14	雷達	軍事	28	293.501	11.333	NOUN
15	裝備	軍事	44	283.992	7.556	NOUN,VERB
16	陸軍	軍事	29	281.022	10.602	NOUN
17	軍事	軍事	80	278.932	4.650	NOUN
18	偵察	軍事	26	272.256	11.333	VERB
19	國防部	軍事	35	257.897	8.440	NOUN
20	裝甲	軍事	24	251.055	11.333	ADJ
21	攻擊	軍事	37	245.925	7.765	VERB
22	指揮	軍事	47	238.443	6.267	NOUN,VERB
23	戰爭	軍事	50	238.094	5.965	NOUN
24	士兵	軍事	29	234.534	9.130	NOUN
25	飛機	軍事	42	231.919	6.704	NOUN
26	彈藥	軍事	23	228.634	10.861	NOUN
27	空中	軍事	29	205.775	8.217	NOUN
28	防務	軍事	23	198.416	9.654	NOUN
29	紅外	軍事	22	197.310	9.973	NOUN
30	飛行員	軍事	20	197.023	10.794	NOUN

Table A.8 Top 30 ranked terms selected in Domain Ontology Graph (醫療)

Rank	Term	Class	Count	x2	R	POS
1	治療	醫療	68	818.958	12.672	VERB
2	病人	醫療	49	529.960	11.650	NOUN
3	藥物	醫療	42	482.650	12.323	ADJ,NOUN
4	醫院	醫療	50	452.105	9.993	NOUN
5	患者	醫療	41	426.179	11.308	NOUN
6	療效	醫療	32	416.038	13.790	NOUN
7	踟躕	醫療	27	350.130	13.790	VERB
8	久遠	醫療	27	350.130	13.790	ADJ
9	尋覓	醫療	27	335.845	13.298	VERB
10	戈壁	醫療	27	335.845	13.298	NOUN
11	荒涼	醫療	27	335.845	13.298	ADJ
12	傳奇	醫療	27	298.542	12.011	ADJ,NOUN
13	皮膚	醫療	26	274.907	11.566	NOUN
14	血壓	醫療	20	258.425	13.790	NOUN
15	血液	醫療	21	232.631	12.066	NOUN
16	疼痛	醫療	20	219.815	11.991	VERB
17	嘔吐	醫療	19	218.593	12.477	VERB
18	服用	醫療	18	218.320	13.064	VERB
19	疾病	醫療	31	217.739	8.221	NOUN
20	血管	醫療	19	207.036	11.910	NOUN
21	出血	醫療	18	205.703	12.411	VERB
22	以免	醫療	25	197.412	9.073	VERB
23	病情	醫療	18	194.294	11.820	NOUN
24	臨床	醫療	18	194.294	11.820	ADJ
25	注射	醫療	17	192.841	12.339	VERB
26	傷口	醫療	17	181.595	11.722	NOUN
27	服藥	醫療	14	180.343	13.790	VERB
28	止血	醫療	14	180.343	13.790	VERB
29	部位	醫療	21	171.348	9.342	NOUN
30	中藥	醫療	15	167.213	12.168	NOUN

Table A.9 Top 30 ranked terms selected in Domain Ontology Graph (電腦)

Rank	Term	Class	Count	x2	R	POS
1	軟件	電腦	89	922.288	10.980	NOUN
2	用戶	電腦	77	865.293	11.874	NOUN
3	程序	電腦	65	486.934	8.460	NOUN
4	計算機	電腦	73	451.662	7.192	NOUN
5	硬盤	電腦	33	442.579	14.187	NOUN
6	操作系統	電腦	34	441.444	13.782	NOUN
7	服務器	電腦	32	414.192	13.757	NOUN
8	微軟	電腦	33	388.310	12.653	NOUN
9	接口	電腦	33	388.310	12.653	NOUN
10	版本	電腦	31	386.746	13.327	CLAS
11	兼容	電腦	31	373.721	12.935	ADJ
12	應用	電腦	66	373.143	6.736	ADJ,VERB
13	計算	電腦	77	372.446	5.905	VERB
14	CPU	電腦	27	360.993	14.187	NOUN
15	內存	電腦	28	359.869	13.698	NOUN
16	硬件	電腦	39	352.166	10.060	NOUN
17	操作	電腦	54	321.717	7.094	VERB
18	NT	電腦	28	321.335	12.414	NOUN
19	IBM	電腦	25	319.286	13.641	NOUN
20	機器	電腦	32	313.124	10.809	NOUN
21	數據	電腦	56	299.626	6.512	NOUN
22	廠商	電腦	28	299.439	11.683	NOUN
23	存儲	電腦	22	293.388	14.187	VERB
24	驅動	電腦	28	279.986	11.034	VERB
25	病毒	電腦	23	266.539	12.550	NOUN
26	系統	電腦	92	266.016	3.991	ADJ,NOUN
27	代碼	電腦	20	251.964	13.511	NOUN
28	編程	電腦	18	239.554	14.187	VERB
29	連接	電腦	30	230.889	8.867	VERB,NOUN
30	電腦	電腦	58	230.627	5.208	NOUN

Table A.10 Top 30 ranked terms selected in Domain Ontology Graph (體育)

Rank	Term	Class	Count	x2	R	POS
1	比賽	體育	236	1017.990	4.860	NOUN
2	冠軍	體育	132	695.166	5.945	NOUN
3	選手	體育	124	660.638	6.018	NOUN
4	決賽	體育	103	523.511	5.862	NOUN
5	女子	體育	97	458.302	5.571	NOUN
6	運動員	體育	87	450.715	5.985	NOUN
7	亞運會	體育	93	447.197	5.653	NOUN
8	亞運	體育	96	441.234	5.464	NOUN
9	金牌	體育	81	391.008	5.698	NOUN
10	錦標賽	體育	72	385.869	6.175	NOUN
11	隊員	體育	84	379.111	5.421	NOUN
12	男子	體育	79	373.759	5.620	ADJ,NOUN
13	奪得	體育	73	352.025	5.713	VERB
14	運動	體育	117	328.558	3.875	NOUN
15	動員	體育	87	300.409	4.501	VERB
16	教練	體育	62	279.354	5.467	NOUN,VERB
17	球隊	體育	50	255.814	6.020	NOUN
18	亞軍	體育	49	250.292	6.015	NOUN
19	參賽	體育	61	243.413	5.025	VERB
20	本屆	體育	65	229.983	4.624	ADJ
21	戰勝	體育	57	226.931	5.026	VERB
22	世界杯	體育	39	209.292	6.260	NOUN
23	奥運	體育	40	200.880	5.962	NOUN
24	球賽	體育	41	199.876	5.833	NOUN
25	奥運會	體育	39	195.422	5.955	NOUN
26	名將	體育	36	192.893	6.260	NOUN
27	參加	體育	155	186.785	2.384	VERB
28	對手	體育	52	179.951	4.585	NOUN
29	擊敗	體育	37	178.141	5.791	VERB
30	預賽	體育	32	171.107	6.260	VERB

Table A.11 Terms dependency in Domain Ontology Graph (文藝)

	戲劇	作品	演唱	創作	文藝	劇院	劇團	演員	精品	表演	文化	藝術	展覽	舞台	觀眾	節目	藝術	演出	舉辦	電影	文化	劇目	晚會	風格	歌舞	舞蹈	美術	歌曲	音樂	戲曲
戲劇	10.82	0.53	0.46	0.95	0.28	1.37	2.76	0.97	0.27	0.28	0.19	0.78	0.38	0.47	0.47	0.19	0.47	0.66	0.16	0.56	0.30	6.38	0.20	0.50	0.75	0.83	0.72	0.10	0.63	1.20
作品	0.40	10.28	0.56	2.77	0.89	0.93	0.78	0.55	1.15	0.16	0.37	1.90	0.83	0.17	0.60	0.25	1.24	0.56	0.55	0.80	0.54	1.48	0.11	0.37	0.59	0.61	2.78	0.55	0.68	2.05
演唱	0.28	0.31	10.39	0.74	1.03	1.33	0.38	1.01	0.00	0.44	0.26	0.79	0.10	0.39	0.80	0.39	0.54	1.37	0.18	0.15	0.37	0.95	0.70	0.13	1.15	1.69	0.12	4.35	1.01	1.98
創作	0.59	1.94	0.75	11.00	0.88	0.79	2.48	0.48	0.91	0.30	0.30	2.19	0.51	0.30	0.51	0.32	1.22	0.80	0.30	0.53	0.55	1.95	0.26	0.40	0.36	0.49	1.66	1.07	0.82	1.62
文藝	0.17	0.60	1.01	0.86	5.24	1.10	1.30	0.82	1.72	0.40	0.33	1.07	0.53	0.09	0.49	0.42	0.91	2.14	0.44	0.24	0.42	1.64	1.43	0.29	0.98	2.08	0.92	0.99	0.55	1.70
劇院	0.73	0.59	1.15	0.68	0.96	10.05	2.81	1.10	0.37	0.47	0.38	0.94	0.24	0.42	1.00	0.45	0.83	1.93	0.25	0.28	0.71	4.33	1.07	0.30	1.38	1.20	0.59	0.61	1.17	3.27
劇團	1.29	0.41	0.29	1.86	1.00	2.45	11.10	1.65	0.00	0.69	0.14	1.33	0.00	0.55	0.69	0.14	0.96	1.92	0.00	0.41	0.65	10.18	0.29	0.00	1.41	1.95	0.64	0.00	0.69	0.00
演員	0.67	0.34	1.15	0.53	0.92	1.42	2.43	10.34	0.36	0.58	0.30	0.74	0.10	0.32	1.00	0.52	0.66	1.32	0.26	0.78	0.52	2.80	0.78	0.33	1.93	1.95	0.64	0.70	0.59	3.71
精品	0.19	0.96	0.00	1.02	1.96	0.48	0.00	0.36	10.20	0.24	0.96	1.00	1.45	0.36	0.72	0.36	1.20	0.24	0.84	0.60	0.29	2.23	0.50	0.84	0.00	0.21	2.79	0.19	0.12	0.00
表演	0.46	0.42	1.33	0.65	1.49	1.05	2.56	1.15	0.55	6.68	0.44	1.05	0.49	0.56	1.15	0.57	0.79	1.44	0.50	0.36	0.51	2.20	1.04	0.28	2.22	2.30	0.66	1.08	0.89	3.04
文化	0.54	1.01	0.69	1.08	1.24	1.02	1.27	0.64	1.28	0.65	5.20	1.15	1.01	0.67	0.90	0.48	1.30	1.20	1.03	0.72	2.40	1.07	0.72	0.57	1.18	1.29	1.27	0.75	0.91	1.85
藝術	0.43	1.11	0.70	1.93	0.97	0.97	1.57	0.59	0.79	0.43	0.38	10.77	0.58	0.32	0.55	0.30	2.13	1.20	0.43	0.38	0.98	1.98	0.54	0.29	0.78	1.22	1.48	0.53	0.78	1.76
展覽	0.35	1.00	0.15	0.75	0.80	0.40	0.00	0.13	1.91	0.46	0.68	0.97	7.97	0.24	0.68	0.18	1.20	0.53	1.36	0.35	0.66	0.00	0.23	0.30	0.63	0.39	1.96	0.26	0.37	0.00
舞台	0.77	0.45	1.26	0.72	0.55	1.20	1.61	1.00	0.47	0.69	0.40	0.80	0.33	8.26	0.77	0.52	0.77	1.19	0.38	0.33	0.39	1.78	0.77	0.56	0.70	1.34	0.51	0.61	0.81	1.58
觀眾	0.46	0.58	1.38	0.82	1.04	1.37	1.97	1.30	0.62	0.75	0.42	0.90	0.58	0.43	8.25	0.73	0.76	1.50	0.49	0.77	0.43	1.66	0.73	0.30	1.35	1.48	0.75	0.91	1.05	2.75
節目	0.49	0.39	1.28	0.62	1.39	0.92	1.11	1.23	0.37	0.66	0.40	0.81	0.27	0.49	1.36	8.83	0.64	1.03	0.36	0.69	0.49	0.80	1.29	0.43	1.26	1.74	0.30	1.12	0.77	2.07
藝術	0.42	1.16	0.65	1.55	1.22	1.08	1.75	0.79	1.31	0.52	0.58	3.31	0.88	0.38	0.71	0.30	8.34	1.26	0.60	0.55	1.18	1.35	0.62	0.40	0.97	1.45	1.90	0.47	0.81	2.03
演出	0.41	0.38	1.56	0.74	2.06	1.98	2.99	1.12	0.25	0.63	0.44	1.28	0.32	0.49	0.98	0.34	1.05	10.15	0.46	0.35	0.83	2.69	0.85	0.16	1.43	1.58	0.54	0.73	1.19	2.98
舉辦	0.31	0.89	1.01	0.76	1.25	0.84	0.53	0.66	1.06	0.57	0.95	1.10	1.63	0.41	0.98	0.38	1.16	1.23	5.93	0.55	1.12	0.40	1.23	0.21	1.32	1.15	1.01	0.99	0.92	1.67
電影	0.69	1.10	0.57	1.18	0.64	0.67	0.96	1.60	0.78	0.35	0.44	0.74	0.52	0.29	1.14	0.52	0.86	0.70	0.42	7.98	0.71	1.20	0.30	0.49	0.71	0.53	1.08	0.83	0.86	0.00
文化	0.23	0.43	0.45	0.67	0.52	1.00	1.06	0.58	0.31	0.54	1.01	1.35	0.55	0.29	0.51	0.40	1.41	1.16	0.76	0.40	10.82	0.67	0.08	0.29	0.62	0.83	0.84	0.23	0.61	5.56
劇目	2.66	0.77	0.64	1.30	1.12	3.36	9.07	1.70	1.33	0.46	0.46	1.49	0.00	0.31	0.77	0.15	0.31	1.70	0.00	0.15	0.37	11.33	0.00	0.31	0.53	1.64	0.71	0.25	0.46	5.93
晚會	0.16	0.10	0.89	0.33	1.85	1.57	0.48	0.89	0.57	0.49	0.33	0.78	0.20	0.26	0.66	0.63	0.66	1.15	0.46	0.16	0.08	0.00	10.20	0.20	2.03	1.75	0.46	0.84	0.73	3.80
風格	0.67	0.65	0.31	1.02	0.86	0.60	0.63	0.62	1.30	0.17	0.39	0.81	0.43	0.54	0.47	0.24	0.65	0.39	0.13	0.54	0.51	1.20	0.32	9.21	0.44	0.46	0.80	0.17	0.52	0.83
歌舞	0.43	0.39	1.06	0.33	0.92	1.47	1.73	1.60	0.00	0.98	0.35	0.81	0.39	0.16	1.02	0.39	0.70	1.49	0.35	0.16	0.47	0.72	1.47	0.12	10.26	2.29	0.72	1.00	0.47	4.51
舞蹈	0.43	0.34	1.40	0.40	1.74	1.15	2.13	1.45	0.16	0.87	0.40	1.13	0.22	0.34	0.78	0.43	0.89	1.49	0.45	0.20	0.56	2.01	1.14	0.14	2.05	9.35	0.67	1.10	1.07	5.57
美術	0.47	1.74	0.12	1.72	0.98	0.71	0.88	0.60	2.58	0.30	0.36	1.74	1.38	0.12	0.42	0.06	1.38	0.72	0.36	0.42	0.71	1.11	0.38	0.30	0.82	0.85	7.82	0.00	0.60	0.00
歌曲	0.07	0.41	5.06	1.24	1.16	0.82	0.00	0.71	0.20	0.39	0.23	0.70	0.21	0.23	0.60	0.34	0.41	0.87	0.25	0.53	0.22	0.42	0.77	0.11	1.26	1.55	0.00	10.72	1.28	1.76
音樂	0.58	0.66	1.93	0.97	0.90	1.56	1.61	0.77	0.17	0.42	0.33	1.14	0.39	0.45	0.89	0.39	0.88	1.60	0.47	0.56	0.57	1.84	0.85	0.32	0.85	1.71	0.78	1.57	7.86	1.91
戲曲	0.42	0.80	1.12	0.90	0.97	2.12	0.00	1.87	0.00	0.54	0.27	1.11	0.00	0.00	0.54	0.27	0.54	1.87	0.00	0.00	2.55	4.96	1.68	0.27	2.75	3.80	0.00	0.85	0.54	11.33

Table A.12 Terms dependency in Domain Ontology Graph (政治)

	雙方	外交	總理	會見	委員	外交	和平	議會	舉行	領導	阿拉	和平	主席	總統	部長	共和	表示	抵達	邊關	外長	外交	關系	巴勒	雙邊	訪問	會談	大使	今天	友好	會晤
雙方	3.02	1.03	0.70	1.25	1.06	0.74	1.70	0.45	0.74	0.78	0.65	0.79	0.71	0.58	0.74	0.92	0.56	0.80	1.91	1.01	0.79	0.92	0.74	1.63	0.93	1.29	0.65	0.45	1.29	1.53
外交	0.49	5.57	0.44	0.91	0.26	1.26	1.21	0.54	0.40	0.42	0.56	0.61	0.42	0.37	1.37	0.75	0.26	0.42	1.19	1.35	2.56	0.54	0.51	0.84	0.68	0.63	0.60	0.32	0.77	0.68
總理	0.88	1.07	3.97	1.12	1.00	0.63	1.48	1.09	0.90	1.22	0.94	0.82	1.25	0.96	0.96	0.80	0.73	0.76	1.25	0.83	0.61	0.76	1.36	0.91	0.84	0.86	0.57	0.57	0.79	0.92
會見	0.80	1.36	0.69	4.39	1.75	0.60	1.75	0.40	0.40	0.65	0.56	0.60	0.61	0.66	0.50	0.65	0.35	1.02	1.91	0.84	0.75	0.67	0.47	1.15	1.12	0.80	0.50	0.31	1.30	1.01
委員	0.38	0.14	0.28	1.15	4.93	0.14	0.70	0.59	0.31	0.42	0.00	0.31	0.45	0.17	0.14	0.56	0.10	0.31	1.01	0.21	0.35	0.59	0.07	0.59	0.31	0.21	0.35	0.24	1.22	1.01
外交	0.89	2.46	0.59	0.92	0.69	4.48	1.38	0.72	0.57	0.74	0.76	0.98	0.70	0.55	0.84	0.87	0.48	0.58	1.46	1.30	2.46	0.85	0.64	1.16	0.88	0.91	0.99	0.57	0.95	0.84
和平	0.42	0.97	0.28	0.63	0.49	0.49	5.35	0.14	0.21	0.35	0.56	0.69	0.35	0.35	0.35	0.76	0.21	0.21	1.46	0.35	0.35	0.42	0.69	0.63	0.21	0.21	0.28	0.35	1.11	0.35
議會	0.55	0.88	1.40	0.76	0.98	1.05	0.57	5.03	1.05	1.08	0.69	0.68	1.32	1.13	0.81	1.03	0.59	0.48	0.76	0.81	0.61	0.96	0.61	0.71	0.59	0.54	0.51	0.45	0.71	0.40
舉行	1.04	1.01	0.92	0.87	1.31	0.81	1.04	0.88	2.05	0.98	0.79	0.88	0.99	0.89	0.91	0.87	0.75	0.94	1.14	1.04	0.77	0.93	0.79	1.12	0.98	1.38	0.84	0.70	1.00	1.38
領導	0.84	0.81	1.10	1.02	1.37	0.70	1.68	0.76	0.84	3.60	1.01	0.91	1.09	0.87	0.63	0.90	0.61	0.70	1.52	0.57	0.55	0.83	1.22	1.06	0.87	0.90	0.49	0.55	0.97	1.59
阿拉	0.72	0.85	0.67	0.74	0.15	0.67	1.79	0.49	0.54	0.97	4.22	1.15	0.75	0.44	0.74	0.25	0.53	0.72	0.56	1.16	0.79	0.59	2.61	0.49	0.98	0.67	0.41	0.62	0.46	0.71
和平	0.82	1.26	0.70	0.93	1.04	0.83	3.55	0.48	0.59	0.87	1.04	3.41	0.70	0.61	0.51	0.92	0.49	0.69	1.39	1.10	0.90	0.62	1.26	1.12	0.77	0.90	0.60	0.67	1.19	1.29
主席	0.84	1.00	1.27	1.10	1.12	0.80	1.40	0.94	0.85	1.15	0.78	0.84	3.31	0.91	0.87	0.98	0.65	0.66	1.17	0.95	0.62	0.89	1.47	0.89	0.85	0.84	0.57	0.69	1.09	0.83
總統	0.89	1.07	1.03	1.30	0.73	0.70	1.11	1.14	0.86	1.08	0.87	0.84	0.96	4.18	0.77	1.11	0.79	1.00	1.21	1.14	0.78	0.93	0.78	0.97	1.17	1.01	0.61	0.61	0.74	1.61
部長	0.85	3.03	0.96	1.00	0.69	0.99	0.87	0.73	0.81	0.75	0.82	0.70	0.88	0.76	2.85	0.85	0.62	0.71	1.03	1.11	1.39	0.79	0.71	1.01	0.83	0.84	0.79	0.59	0.90	0.88
共和	0.62	1.34	0.41	0.67	1.12	0.53	1.81	0.55	0.53	0.53	0.17	0.24	0.41	0.60	0.55	2.92	0.41	0.43	1.08	0.72	0.69	0.55	0.00	0.79	0.57	0.33	0.65	0.33	1.39	0.50
表示	1.04	0.98	0.95	1.12	1.03	0.99	1.04	0.89	0.91	0.96	0.91	1.01	0.94	0.91	0.91	0.94	2.32	0.91	1.06	1.12	1.04	0.92	0.85	1.02	1.04	1.16	0.98	0.79	1.05	1.02
抵達	0.88	0.53	0.54	1.52	1.24	0.42	0.72	0.39	0.68	0.65	0.73	0.68	0.46	0.54	0.43	0.39	0.36	4.13	1.38	0.59	0.56	0.60	0.67	1.02	1.66	1.38	0.56	0.36	0.85	1.32
邊關	0.68	1.27	0.39	0.96	0.89	0.50	2.06	0.23	0.35	0.46	0.21	0.27	0.50	0.44	0.37	0.75	0.23	0.44	5.38	0.42	0.68	0.89	0.17	4.21	0.64	0.60	0.39	0.17	1.04	0.71
外長	0.88	1.89	0.60	0.94	0.32	0.92	1.71	0.51	0.60	0.59	0.98	0.96	0.77	0.61	0.66	0.76	0.43	0.45	1.29	5.07	1.22	0.57	0.94	0.87	0.84	1.33	0.62	0.49	0.71	0.83
外交	0.79	4.17	0.49	0.86	0.57	2.00	1.17	0.47	0.47	0.53	0.55	0.81	0.46	0.39	1.05	0.88	0.41	0.48	1.27	1.33	5.21	0.73	0.43	1.03	0.82	0.78	1.01	0.44	0.84	0.69
關系	1.00	1.15	0.64	1.08	1.28	0.88	1.81	0.68	0.62	0.86	0.67	0.77	0.76	0.67	0.67	0.93	0.50	0.69	2.40	0.92	0.95	2.57	0.51	1.77	0.90	0.80	0.83	0.57	1.69	1.04
巴勒	0.93	0.71	1.64	0.53	0.04	0.78	2.04	0.49	0.66	1.50	2.35	1.60	1.80	0.69	0.84	0.05	0.60	0.56	0.27	1.04	0.57	0.61	4.77	0.22	0.83	1.02	0.19	0.56	0.22	0.76
雙邊	0.98	1.21	0.36	1.02	0.85	0.58	1.68	0.30	0.46	0.52	0.24	0.46	0.53	0.53	0.48	0.66	0.41	0.55	5.78	0.59	0.73	0.88	0.16	4.97	0.69	0.91	0.58	0.28	1.13	1.00
訪問	0.90	1.06	0.66	1.39	1.37	0.79	1.30	0.38	0.71	0.77	0.83	0.73	0.68	0.85	0.68	0.79	0.56	1.45	1.78	0.89	0.78	0.80	0.69	1.21	3.72	1.48	0.54	0.40	1.18	1.17
會談	1.27	0.90	0.73	1.14	1.19	0.73	1.06	0.35	0.93	0.84	0.64	0.86	0.68	0.71	0.65	0.59	0.62	1.16	1.40	1.18	0.83	0.76	0.87	1.60	1.54	4.50	0.55	0.49	0.52	1.66
大使	0.85	0.97	0.57	0.86	1.12	1.06	1.40	0.51	0.61	0.53	0.53	0.68	0.61	0.48	0.71	1.15	0.57	0.76	1.30	0.86	1.11	0.92	0.23	1.00	0.68	0.69	4.02	0.73	1.76	0.66
今天	0.58	0.92	0.69	0.91	0.83	0.82	1.70	0.51	0.74	0.71	0.91	0.86	0.96	0.54	0.79	0.83	0.53	0.69	0.99	1.04	0.65	0.76	0.78	0.84	0.59	0.73	0.88	1.91	1.07	0.94
友好	0.85	1.18	0.45	1.40	2.10	0.60	2.92	0.36	0.37	0.48	0.37	0.57	0.57	0.29	0.50	1.18	0.33	0.63	2.13	0.73	0.71	1.07	0.16	1.35	0.73	0.41	1.11	0.54	4.04	0.84
會晤	0.99	1.00	0.61	1.12	0.88	0.55	0.89	0.29	0.81	1.07	0.60	0.71	0.65	0.85	0.44	0.59	0.59	0.92	1.65	0.88	0.57	0.76	0.77	1.37	0.99	1.17	0.40	0.27	0.91	4.76

Table A.13 Terms dependency in Domain Ontology Graph (交通)

	行駛	客運	路局	運量	駕駛	仁慈	鐵道	列車	鐵道	客車	旅客	車站	公交	鐵路	運輸	駕駛	通行	公安	通車	車輛	堅韌	星期	交通	不忍	交通	公安	公安	貨運	違章	公路
行駛	11.33	0.46	1.53	0.26	0.92	0.00	1.30	0.99	0.99	0.82	0.23	0.44	0.95	0.43	0.24	0.56	0.25	0.00	1.00	1.06	0.00	0.02	0.50	0.10	0.37	0.30	0.18	0.30	1.76	0.82
客運	0.31	11.79	0.00	9.20	0.44	0.00	1.31	2.28	1.37	1.00	1.21	0.69	0.40	1.59	0.57	0.04	0.23	0.00	0.95	0.12	0.00	0.06	0.72	0.00	0.16	0.00	0.00	2.13	0.00	0.20
路局	0.89	0.00	13.15	0.00	0.00	0.00	0.00	1.96	0.00	0.36	1.13	0.65	0.40	2.05	0.82	0.00	0.00	0.00	1.60	0.20	0.00	0.00	0.91	0.00	1.02	0.00	0.00	0.38	0.00	1.43
運量	0.16	8.56	0.00	13.15	0.00	0.00	0.00	0.81	1.66	0.00	0.98	0.78	0.29	1.04	0.74	0.00	0.42	0.00	2.31	0.00	0.00	0.00	0.00	0.00	0.30	0.00	0.00	4.97	0.00	0.15
駕駛	0.89	0.64	0.00	0.00	11.12	0.00	0.62	0.47	0.22	0.54	0.34	0.24	0.15	0.15	0.23	3.55	0.11	0.00	0.00	0.35	0.00	0.17	0.52	0.00	0.27	0.10	0.21	0.29	1.88	0.15
仁慈	0.00	0.00	0.00	0.00	0.00	12.62	0.00	0.00	0.00	0.00	0.46	0.00	0.00	0.42	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.11	0.00	0.00	0.00	0.42
鐵道	0.78	1.18	0.00	0.00	0.39	0.00	11.76	1.36	24.80	0.50	0.63	1.35	0.28	0.57	0.00	0.14	0.00	0.00	1.11	0.43	0.00	0.00	0.63	0.00	0.00	0.00	0.00	0.27	0.00	0.00
列車	0.76	2.61	2.57	1.00	0.38	0.00	1.74	11.87	2.08	0.41	0.65	1.51	0.67	1.70	0.37	0.15	0.35	0.00	2.05	0.22	0.00	0.05	0.31	0.00	0.20	0.04	0.09	1.87	0.00	0.25
鐵道	0.70	1.44	0.00	1.87	0.16	0.00	28.97	1.90	11.20	0.30	0.51	1.16	0.46	1.33	0.12	0.17	0.33	0.00	1.81	0.17	0.00	0.00	0.39	0.00	0.29	0.00	0.00	0.86	0.00	0.29
客車	0.94	1.71	0.70	0.00	0.64	0.00	0.95	0.61	0.50	11.74	0.42	0.28	0.47	0.41	0.30	0.56	0.42	0.00	0.93	0.38	0.00	0.00	0.72	0.00	0.33	0.00	0.00	0.39	0.72	0.50
旅客	0.28	2.19	2.33	1.90	0.43	0.31	1.26	1.02	0.88	0.45	12.68	0.95	0.23	0.53	0.31	0.16	0.17	0.00	1.07	0.12	0.00	0.06	0.52	0.12	0.27	0.00	0.00	0.73	0.48	0.27
車站	0.45	1.04	1.12	1.27	0.26	0.00	2.28	2.00	1.68	0.25	0.80	10.52	1.49	0.71	0.24	0.17	0.27	0.00	1.35	0.33	0.29	0.07	0.21	0.10	0.46	0.21	0.35	0.76	0.00	0.09
公交	0.96	0.60	0.70	0.47	0.16	0.00	0.47	0.88	0.66	0.41	0.19	1.48	11.91	0.85	0.26	0.32	0.38	0.00	0.46	0.73	0.00	0.04	0.46	0.18	0.56	0.16	0.00	0.27	0.71	0.23
鐵路	0.39	2.40	3.77	1.53	0.11	0.16	2.55	2.26	2.55	0.39	0.50	0.91	0.81	9.55	0.68	0.17	0.70	0.00	2.23	0.23	0.39	0.03	0.45	0.00	0.43	0.08	0.12	1.44	0.13	0.85
運輸	0.64	1.51	1.99	2.19	0.80	0.16	1.35	0.93	1.00	0.77	0.64	0.56	0.58	1.38	8.49	0.43	0.48	0.00	1.14	0.42	0.77	0.06	0.86	0.33	0.68	0.25	0.41	1.62	0.51	0.47
駕駛	1.05	0.28	0.32	0.00	6.68	0.00	0.22	0.44	0.23	0.87	0.32	0.23	0.48	0.24	0.23	8.34	0.25	0.00	0.42	0.67	0.00	0.14	0.59	0.17	0.37	0.36	0.45	0.20	2.61	0.49
通行	0.31	0.42	0.00	0.83	0.14	0.00	0.00	0.56	0.58	0.45	0.17	0.32	0.46	0.46	0.21	0.23	9.07	0.00	1.40	0.97	0.00	0.19	0.91	0.16	0.31	0.21	0.14	0.81	0.94	0.51
公安	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	12.37	0.00	0.00	0.00	0.00	0.00	0.00	0.00	16.83	13.11	0.00	0.00	0.00
通車	0.68	0.97	1.85	2.51	0.00	0.00	1.26	1.81	1.75	0.55	0.60	0.90	0.31	1.64	0.31	0.31	0.78	0.00	11.95	0.47	0.00	0.00	1.04	0.00	0.31	0.00	0.00	1.16	0.95	0.55
車輛	1.49	0.41	0.30	0.40	0.65	0.00	0.81	0.55	0.28	0.64	0.23	0.53	1.14	0.31	0.26	0.65	1.51	0.00	1.76	8.89	0.23	0.04	0.75	0.16	0.59	0.30	0.14	0.42	1.98	0.79
堅韌	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.51	0.00	0.96	0.00	0.00	0.00	0.00	0.00	0.48	12.62	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
星期	0.04	0.16	0.00	0.00	0.32	0.00	0.00	0.11	0.00	0.00	0.09	0.12	0.08	0.08	0.12	0.20	0.28	0.00	0.00	0.08	0.00	10.52	0.00	0.24	0.16	0.21	0.43	0.07	0.00	0.16
交通	0.50	1.08	1.56	0.00	0.53	0.00	1.06	0.40	0.55	0.63	0.43	0.21	0.45	0.52	0.26	0.20	0.75	0.00	1.54	0.56	0.00	0.00	12.60	0.41	1.34	0.18	0.36	0.55	2.40	0.66
不忍	0.18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.18	0.17	0.32	0.00	0.16	0.00	0.23	0.00	0.00	0.16	0.00	0.24	0.71	13.15	0.00	0.86	0.89	0.00	0.00	0.00
交通	0.94	0.94	0.91	0.72	0.71	0.20	0.51	0.88	0.86	0.87	0.68	0.74	1.61	1.00	0.67	0.50	0.59	0.00	1.69	0.83	0.24	0.13	3.18	0.28	5.18	0.46	0.32	0.66	1.70	0.94
公安	0.35	0.00	0.00	0.00	0.12	0.71	0.00	0.06	0.00	0.00	0.00	0.24	0.18	0.09	0.09	0.23	0.20	22.72	0.00	0.23	0.00	0.13	0.20	0.57	0.14	7.60	22.72	0.00	0.55	0.00
公安	0.16	0.00	0.00	0.00	0.20	0.00	0.00	0.10	0.00	0.00	0.00	0.31	0.00	0.07	0.00	0.22	0.11	13.88	0.00	0.00	0.00	0.22	0.33	0.46	0.00	17.81	9.44	0.00	0.90	0.00
貨運	0.27	2.89	0.59	7.25	0.27	0.00	0.40	2.22	1.12	0.31	0.55	0.68	0.25	1.30	0.50	0.10	0.61	0.00	1.56	0.17	0.00	0.04	0.50	0.00	0.25	0.00	0.00	11.20	0.00	0.25
違章	1.55	0.00	0.00	0.00	1.71	0.00	0.00	0.00	0.00	0.55	0.35	0.00	0.62	0.16	0.16	1.10	0.68	0.00	1.23	1.10	0.00	0.00	2.10	0.00	0.47	0.42	0.87	0.00	11.60	0.63
公路	1.34	0.60	1.98	0.50	0.57	0.33	0.34	0.57	0.64	0.86	0.51	0.32	0.68	1.33	0.41	0.55	0.58	0.00	2.19	0.94	0.19	0.14	0.97	0.20	0.96	0.17	0.00	0.50	1.40	8.34

Table A.14 Terms dependency in Domain Ontology Graph (教育)

	課堂	素質	教材	家長	學習	辦學	畢業	校園	德育	教育	教師	初中	小學	培養	師生	學校	學科	高等	教學	家教	學生	高中	課程	教職	教委	校長	老師	師資	中學	大學
課堂	12.03	0.27	0.84	1.04	0.63	0.60	0.37	0.70	0.00	0.26	1.11	0.51	1.00	0.32	1.39	0.53	0.40	0.47	1.76	0.56	0.42	0.47	2.01	0.00	0.00	0.85	1.77	1.12	0.84	0.21
素質	0.66	7.06	0.40	0.34	0.37	0.00	0.34	0.60	0.00	0.52	0.37	0.36	0.37	1.05	0.08	0.30	0.58	0.56	0.47	0.00	0.19	0.45	0.69	0.00	1.25	0.56	0.57	0.80	0.22	0.19
教材	1.29	0.25	11.20	0.10	0.58	4.15	0.22	0.20	0.00	0.44	1.02	0.70	0.51	0.73	0.32	0.37	1.41	0.22	1.85	1.55	0.66	0.37	1.79	0.00	0.00	0.93	0.48	2.33	0.80	0.22
家長	1.69	0.23	0.10	9.85	0.44	1.52	0.35	0.82	0.00	0.44	0.88	1.11	0.84	0.31	0.80	0.81	0.29	0.31	0.66	0.61	0.73	0.81	0.53	0.00	1.92	0.69	1.39	1.02	0.59	0.33
學習	1.95	0.54	1.03	0.65	5.79	1.47	0.90	0.49	0.00	0.58	0.83	0.90	0.84	0.83	0.94	0.70	1.00	0.53	1.26	0.61	0.85	0.68	1.71	1.20	0.48	0.86	1.19	1.46	1.19	0.50
辦學	0.59	0.00	2.66	0.92	0.50	12.20	0.70	0.13	0.00	0.60	0.90	0.96	0.60	1.00	0.22	0.70	0.38	0.50	1.50	1.06	0.60	0.50	1.83	0.00	6.67	1.62	0.76	6.38	0.40	0.30
畢業	0.70	0.46	0.40	0.61	0.82	1.48	7.91	0.81	0.00	0.38	0.82	0.83	0.52	0.66	0.81	0.55	0.46	0.64	0.66	1.15	0.59	1.18	1.19	0.00	0.72	0.85	1.05	2.19	0.97	0.71
校園	1.04	0.36	0.19	0.74	0.20	0.20	0.41	9.96	0.00	0.27	0.90	0.68	0.48	0.35	1.91	1.21	0.34	0.35	0.76	0.19	0.64	0.96	0.73	2.96	0.59	1.41	1.06	0.75	0.69	0.34
德育	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	12.18	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
教育	1.31	0.73	1.21	0.90	0.86	1.56	0.69	0.70	0.00	4.57	1.04	0.99	1.04	0.90	0.70	0.81	0.96	1.19	1.18	0.90	0.84	0.85	1.09	0.88	1.41	0.95	0.99	1.69	0.84	0.53
教師	1.92	0.21	1.22	1.04	0.46	1.61	0.56	0.87	0.00	0.38	10.47	0.79	0.74	0.50	1.20	0.72	0.50	0.33	1.24	1.16	0.60	0.76	1.05	1.82	1.45	0.94	1.35	2.55	0.89	0.27
初中	0.80	0.23	0.72	1.07	0.48	1.55	0.41	0.73	0.00	0.34	0.68	10.13	1.36	0.14	0.00	0.41	0.26	0.07	0.86	0.72	0.48	1.77	0.83	0.00	4.54	0.39	0.74	0.00	1.16	0.14
小學	1.81	0.22	0.68	0.96	0.63	1.02	0.43	0.60	0.00	0.45	0.83	1.97	8.49	0.46	0.56	0.67	0.39	0.32	0.99	0.95	0.79	0.82	0.96	0.00	2.13	0.56	1.22	0.95	1.07	0.19
培養	0.86	0.86	1.03	0.38	0.68	1.84	0.58	0.45	0.00	0.49	0.68	0.16	0.55	7.49	0.57	0.70	1.00	0.49	0.93	0.52	0.47	0.31	1.09	0.00	1.62	0.80	0.53	1.72	0.50	0.31
師生	1.96	0.05	0.30	0.69	0.31	0.32	0.45	1.82	0.00	0.19	1.03	0.00	0.39	0.42	10.06	0.72	0.75	0.31	1.06	0.30	0.53	0.64	0.55	0.00	0.00	2.25	1.24	0.30	0.61	0.14
學校	1.49	0.33	0.74	1.00	0.59	1.58	0.61	1.61	0.00	0.51	1.04	0.95	1.03	0.58	1.29	8.63	0.71	0.56	1.31	1.06	0.82	0.83	1.25	2.48	0.99	1.87	1.26	1.64	0.90	0.35
學科	0.67	0.39	1.50	0.30	0.62	0.64	0.34	0.38	0.00	0.17	0.68	0.27	0.34	0.79	0.87	0.68	8.54	0.68	1.30	0.60	0.62	0.51	1.47	0.00	0.00	0.65	0.67	2.41	0.62	0.40
高等	1.02	0.51	0.46	0.46	0.45	1.58	0.64	0.58	0.00	0.78	0.43	0.08	0.42	0.56	0.54	0.64	0.87	8.27	0.68	0.74	0.61	0.43	1.04	0.00	0.58	0.82	0.41	1.48	0.40	0.40
教學	2.33	0.26	1.60	0.54	0.65	2.01	0.33	0.67	0.00	0.46	1.05	0.72	0.72	0.60	0.99	0.78	1.05	0.38	10.86	1.45	0.70	0.48	1.44	0.00	1.36	1.06	0.93	2.47	0.68	0.31
家教	0.84	0.00	1.52	0.56	0.43	1.62	0.29	0.19	0.00	0.00	1.29	0.69	0.71	0.43	0.31	0.14	0.55	0.57	1.65	11.46	0.00	0.86	1.53	0.00	4.76	1.15	0.62	6.08	1.29	0.00
學生	1.42	0.33	0.89	0.91	0.78	1.14	0.74	1.47	0.00	0.49	1.05	0.94	1.26	0.67	1.40	0.92	0.80	0.55	1.15	0.71	7.61	1.00	1.11	1.39	0.83	1.20	1.23	1.42	1.28	0.50
高中	0.75	0.31	0.38	0.80	0.44	0.80	0.78	1.06	0.00	0.48	0.76	1.77	0.73	0.27	0.75	0.69	0.49	0.31	0.57	0.90	0.73	10.59	0.61	0.00	0.94	0.62	1.03	0.90	1.09	0.38
課程	2.56	0.36	1.48	0.41	0.83	2.37	0.57	0.63	0.00	0.38	0.94	0.67	0.71	0.64	0.50	0.70	1.14	0.47	1.38	1.29	0.64	0.49	11.32	0.00	0.58	0.88	0.92	2.22	0.76	0.43
教職	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.23	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.17	0.00	0.00	0.00	0.00	0.00	0.00	0.00	12.81	0.00	0.00	0.00	0.00	3.17	0.00
教委	0.00	0.58	0.00	1.34	0.34	7.71	0.34	0.45	0.00	0.68	1.02	3.26	1.36	1.02	0.00	1.02	0.00	0.34	1.17	3.61	0.34	0.68	0.52	0.00	10.98	1.18	0.00	3.61	0.68	0.00
校長	1.21	0.33	0.86	0.60	0.37	2.34	0.35	1.35	0.00	0.32	0.78	0.35	0.43	0.53	2.26	1.07	0.56	0.41	1.13	1.09	0.68	0.54	0.99	0.00	1.47	11.72	1.07	2.34	0.44	0.26
老師	2.36	0.31	0.41	1.14	0.56	1.03	0.61	0.95	0.00	0.43	1.07	0.62	0.91	0.35	1.17	0.73	0.55	0.23	0.94	0.55	0.56	0.82	0.97	0.00	0.00	1.00	9.74	1.65	0.87	0.25
師資	0.97	0.28	1.32	0.54	0.58	5.63	0.74	0.44	0.00	0.41	1.49	0.00	0.50	0.83	0.18	0.58	1.27	0.58	1.62	3.51	0.50	0.50	1.52	0.00	2.75	1.43	1.07	12.81	0.83	0.33
中學	1.59	0.14	0.96	0.77	0.72	1.02	0.63	1.04	0.00	0.29	1.09	1.25	0.98	0.43	0.92	0.65	0.52	0.28	0.90	1.32	0.70	1.17	1.02	1.88	1.13	0.64	1.23	1.32	9.68	0.19
大學	0.98	0.47	0.75	0.57	0.89	1.23	1.58	1.30	0.00	0.57	0.88	0.74	0.67	0.78	1.38	0.76	1.28	0.78	0.93	0.89	1.02	0.89	1.18	0.69	0.69	1.56	0.99	1.42	0.78	5.30

Table A.15 Terms dependency in Domain Ontology Graph (環境)

	污水	植物	環保	資源	野生	排放	人類	野生	水源	土壤	綠色	污染	水污	廢水	污染	環境	流域	垃圾	自然	回收	環保	地球	大氣	生態	防治	水質	污染	動物	保護	森林
污水	12.99	0.56	0.83	0.36	0.00	1.19	0.24	0.00	2.35	0.58	0.42	1.55	7.73	7.45	1.15	0.42	1.46	1.43	0.18	1.45	0.00	0.30	0.16	0.60	0.55	4.01	2.33	0.24	0.30	0.18
植物	0.68	8.07	0.51	0.32	1.76	0.35	0.30	3.50	0.71	0.98	0.51	0.62	0.00	1.13	0.00	0.62	0.80	0.32	0.41	0.33	0.00	0.65	0.58	0.81	0.15	1.82	1.27	1.19	0.38	0.54
環保	1.03	0.72	11.01	0.44	0.54	1.51	0.50	0.50	0.63	0.35	1.49	1.24	2.10	1.86	1.59	0.70	0.46	0.65	0.46	1.43	5.16	0.63	0.43	1.16	0.33	1.88	1.69	0.31	0.42	0.40
資源	0.97	0.72	0.85	4.16	0.73	0.80	0.70	0.61	1.63	1.09	0.58	0.90	2.46	1.60	1.02	0.76	1.30	0.48	1.18	0.91	1.66	0.70	0.56	1.25	0.64	1.36	1.31	0.42	0.63	0.74
野生	0.00	1.24	0.31	0.26	13.99	0.04	0.39	15.17	0.38	0.21	0.09	0.18	0.00	0.00	0.00	0.22	0.00	0.10	0.44	0.00	0.00	0.13	0.12	0.53	0.08	0.00	0.00	3.46	0.31	0.70
排放	1.66	0.41	1.31	0.37	0.15	11.36	0.69	0.28	0.70	0.42	1.13	1.17	2.33	2.55	1.76	0.63	0.39	0.46	0.34	1.06	2.15	0.84	0.78	1.08	0.23	1.21	3.22	0.19	0.32	0.74
人類	0.70	0.84	0.66	0.38	1.22	1.02	5.43	1.19	0.95	0.91	0.60	0.89	1.39	1.43	0.72	0.53	0.83	0.42	0.64	0.48	0.59	1.50	0.97	1.04	0.51	0.81	1.25	1.20	0.38	0.57
野生	0.00	2.62	0.29	0.29	16.08	0.16	0.39	13.99	0.56	0.23	0.23	0.29	0.00	0.89	0.00	0.29	0.16	0.11	0.42	0.08	0.00	0.23	0.13	0.48	0.06	0.62	0.25	3.09	0.29	0.64
水源	2.51	0.63	0.39	0.78	0.48	0.50	0.50	0.67	12.12	1.76	0.17	1.67	4.13	3.10	1.07	0.61	0.82	0.43	0.17	0.27	0.00	0.56	0.52	0.95	0.31	5.36	2.61	0.17	0.28	0.56
土壤	0.78	1.10	0.47	0.58	0.34	0.43	0.43	0.35	2.21	9.12	0.31	0.82	1.44	1.62	0.75	0.70	0.38	0.25	0.78	0.47	0.00	0.66	0.62	0.70	0.36	0.37	2.12	0.19	0.39	0.58
綠色	0.67	0.79	1.07	0.15	0.15	1.05	0.40	0.40	0.22	0.41	6.83	0.74	1.24	0.93	0.96	0.42	0.25	0.57	0.45	1.30	0.00	0.40	0.38	1.10	0.31	0.16	1.44	0.30	0.27	0.60
污染	1.89	0.65	0.93	0.35	0.38	1.00	0.52	0.49	2.18	0.71	0.66	9.86	5.40	2.40	5.40	0.68	0.58	0.60	0.29	1.11	2.70	0.54	0.80	1.19	0.38	2.81	5.40	0.38	0.29	0.48
水污	4.54	0.00	0.91	0.39	0.00	1.17	0.26	0.00	2.28	0.63	0.26	2.47	13.99	3.61	2.50	0.65	1.27	0.71	0.39	0.95	0.00	0.26	0.69	1.04	1.21	4.99	6.09	0.00	0.52	0.52
廢水	6.20	0.78	0.96	0.41	0.00	1.79	0.28	0.83	2.42	1.00	0.41	1.79	5.10	13.25	2.65	0.28	0.00	0.60	0.14	0.67	0.00	0.28	0.37	0.41	0.26	6.62	5.38	0.28	0.14	0.14
污染	1.00	0.00	0.60	0.60	0.00	0.80	0.40	0.00	0.88	0.48	0.60	3.60	3.70	2.77	13.99	0.80	0.00	0.44	0.60	0.49	12.48	0.00	0.00	0.80	0.37	3.84	7.80	0.00	0.60	0.20
環境	0.93	0.84	1.23	0.69	0.61	1.37	0.79	0.73	1.00	0.84	0.96	1.42	2.01	1.08	1.39	3.31	0.79	0.61	0.70	0.91	2.26	0.80	0.67	1.58	0.46	1.24	1.66	0.56	0.67	0.76
流域	2.03	0.92	0.41	0.81	0.00	0.32	0.49	0.24	1.07	0.39	0.08	0.49	3.01	0.00	0.00	0.49	10.88	0.18	0.41	0.20	0.00	0.57	0.76	0.57	1.36	3.12	0.63	0.57	0.65	0.57
垃圾	2.38	0.45	0.58	0.11	0.19	0.50	0.32	0.20	0.66	0.32	0.58	0.78	2.00	1.20	0.83	0.43	0.21	9.32	0.22	2.16	0.00	0.37	0.58	0.48	0.16	1.66	1.35	0.17	0.22	0.37
自然	0.55	1.09	0.66	1.02	1.62	0.82	0.90	1.48	0.74	0.99	0.75	0.75	1.57	0.35	0.82	0.71	0.83	0.39	4.87	0.41	0.53	0.77	0.56	1.27	0.36	0.73	0.86	0.97	0.64	0.88
回收	1.94	0.37	1.05	0.27	0.00	0.82	0.27	0.12	0.34	0.47	1.13	1.17	2.16	1.08	0.75	0.39	0.19	1.74	0.12	9.32	0.00	0.39	0.26	0.82	0.22	1.49	0.61	0.08	0.19	0.23
環保	0.00	0.00	2.60	0.00	0.00	1.73	0.00	0.00	0.00	0.00	0.00	0.87	0.00	0.00	16.64	0.87	0.00	0.00	0.87	0.00	13.16	0.00	0.00	0.87	0.00	8.32	6.76	0.00	0.87	0.00
地球	0.64	0.83	0.59	0.41	0.28	1.04	1.16	0.46	0.90	1.03	0.45	0.72	1.19	1.25	0.00	0.46	0.63	0.49	0.42	0.66	0.00	7.99	2.34	0.91	0.29	0.74	1.31	0.33	0.18	0.48
大氣	0.24	0.71	0.24	0.21	0.20	0.73	0.47	0.21	0.72	0.68	0.28	0.99	1.74	0.65	0.00	0.42	0.81	0.51	0.26	0.29	0.00	1.67	9.32	0.66	0.31	1.36	2.94	0.16	0.24	0.45
生態	0.75	0.95	0.76	0.38	0.83	0.94	0.58	0.69	1.19	0.72	0.84	1.08	2.26	0.56	1.04	0.68	0.80	0.38	0.45	0.73	0.85	0.62	0.58	10.39	0.20	1.56	1.91	0.28	0.45	0.70
防治	1.08	0.25	0.30	0.35	0.19	0.30	0.22	0.13	0.57	0.52	0.35	0.56	4.00	0.60	0.83	0.39	1.90	0.19	0.30	0.32	0.00	0.26	0.40	0.26	6.68	0.83	1.01	0.30	0.30	0.39
水質	3.23	1.22	1.20	0.18	0.00	0.65	0.28	0.56	4.05	0.22	0.00	1.85	6.83	6.41	3.55	0.46	1.81	0.81	0.18	0.90	5.77	0.37	0.74	0.92	0.34	13.99	5.05	0.37	0.18	0.28
污染	1.63	0.74	0.59	0.26	0.00	1.43	0.52	0.20	1.71	1.11	0.52	2.67	7.24	4.52	6.26	0.46	0.32	0.57	0.33	0.32	4.07	0.52	1.39	0.98	0.36	4.38	12.77	0.07	0.20	0.33
動物	0.30	1.60	0.32	0.20	7.54	0.21	0.83	6.04	0.26	0.22	0.29	0.41	0.00	0.42	0.29	0.41	0.52	0.20	0.45	0.07	0.00	0.29	0.20	0.44	0.56	0.58	0.12	7.30	0.42	0.69
保護	0.69	0.85	1.02	0.74	1.60	0.89	0.65	1.51	0.72	0.67	0.89	0.93	2.19	0.91	1.14	0.90	0.90	0.53	0.78	0.75	1.23	0.60	0.67	1.34	0.62	1.39	1.44	0.96	5.09	0.89
森林	0.35	0.77	0.41	0.27	1.31	0.82	0.34	1.23	0.86	0.77	0.60	0.62	1.31	0.25	0.34	0.37	0.70	0.35	0.35	0.26	0.00	0.35	0.50	0.90	0.33	0.51	0.83	0.78	0.41	11.11

Table A.16 Terms dependency in Domain Ontology Graph (經濟)

	增長	價格	收入	商品	財政	生產	貿易	幅度	金融	總額	大幅	產品	貨幣	消費	增長	出口	銀行	宏觀	企業	投資	通貨	同期	美元	資本	經濟	百分	市場	增長	季度	下降
增長	5.15	0.67	0.76	0.94	0.53	0.65	0.73	1.06	0.66	0.82	0.71	0.64	1.14	0.89	2.46	0.78	0.52	1.67	0.62	0.80	1.44	1.25	0.38	0.96	1.20	0.46	0.70	2.46	1.37	0.82
價格	0.63	4.99	0.66	1.46	0.25	0.68	0.40	1.26	0.61	0.48	0.91	0.96	1.28	1.24	1.18	0.73	0.47	1.14	0.55	0.58	2.08	1.20	0.58	1.04	1.08	0.39	1.04	1.19	1.54	0.71
收入	0.88	0.79	5.87	0.74	0.60	0.60	0.29	0.97	0.56	0.85	0.63	0.57	1.14	0.83	1.31	0.65	0.73	1.24	0.66	0.79	1.26	1.07	0.63	1.02	1.27	0.61	0.57	1.48	1.10	0.76
商品	0.73	1.23	0.34	5.50	0.19	0.77	0.82	0.78	0.49	0.61	0.53	1.27	1.09	1.38	1.17	1.17	0.33	0.88	0.62	0.54	1.32	1.01	0.46	1.05	0.83	0.25	0.93	1.03	0.98	0.57
財政	0.63	0.39	0.76	0.62	5.82	0.35	0.50	0.98	1.14	0.83	0.51	0.41	1.45	0.52	1.55	0.51	0.95	2.53	0.40	0.46	1.45	0.94	0.47	1.14	1.30	0.56	0.38	1.26	1.62	0.63
生產	0.95	1.01	0.84	1.40	0.39	3.41	0.59	0.84	0.47	0.79	0.58	1.43	0.80	1.51	1.51	1.26	0.44	1.12	0.86	0.66	1.23	0.96	0.56	0.98	0.81	0.38	0.84	1.23	1.27	0.66
貿易	1.24	0.61	0.47	1.49	0.53	0.70	4.93	0.69	0.88	1.02	0.47	1.31	1.32	0.96	1.21	1.32	0.55	1.81	0.99	1.18	0.92	0.82	0.50	1.18	0.82	0.42	0.86	1.05	0.68	0.50
幅度	0.48	0.61	0.38	0.56	0.27	0.35	0.21	5.53	0.33	0.58	2.03	0.38	1.05	0.75	0.96	0.33	0.23	1.21	0.36	0.29	1.59	1.11	0.13	0.96	1.09	0.63	0.35	1.05	1.46	0.73
金融	0.63	0.66	0.56	0.78	0.81	0.32	0.58	0.89	6.64	0.55	0.56	0.50	1.77	0.66	1.57	0.43	1.09	2.63	0.54	0.84	1.36	1.12	0.44	1.44	1.38	0.50	0.67	1.67	1.73	0.45
總額	0.92	0.40	0.71	0.75	0.58	0.58	0.62	0.62	0.46	6.99	0.75	0.65	0.50	0.60	0.81	0.98	0.62	0.99	0.73	0.73	0.52	1.06	0.98	0.67	0.52	0.33	0.50	1.11	1.16	0.56
大幅	0.71	1.00	0.55	0.82	0.37	0.41	0.34	3.08	0.55	0.76	5.65	0.54	1.07	0.79	1.19	0.60	0.39	1.08	0.45	0.45	1.43	1.40	0.31	0.77	0.98	0.65	0.57	1.25	1.61	1.01
產品	0.87	1.18	0.60	1.80	0.34	1.22	1.10	0.67	0.61	0.69	0.57	3.90	1.00	1.73	0.80	1.67	0.56	1.05	0.98	0.69	0.90	0.90	0.35	0.95	0.86	0.31	1.00	0.91	0.85	0.46
貨幣	0.65	0.63	0.55	0.93	0.53	0.26	0.53	1.02	0.92	0.34	0.53	0.50	6.82	0.73	1.51	0.53	1.04	2.71	0.33	0.61	1.91	0.71	0.30	1.46	1.15	0.24	0.68	1.59	1.34	0.50
消費	0.70	1.05	0.55	1.56	0.27	0.91	0.60	1.09	0.46	0.58	0.61	1.18	1.07	5.52	1.38	0.85	0.45	1.38	0.68	0.57	1.53	1.22	0.30	1.06	1.18	0.33	0.76	1.38	1.49	0.69
增長	1.04	0.39	0.52	0.80	0.41	0.54	0.28	0.69	0.43	0.37	0.58	0.35	1.28	0.63	8.19	0.58	0.41	1.42	0.41	0.45	1.78	0.99	0.09	0.69	1.13	0.19	0.37	2.22	1.47	0.65
出口	0.98	0.82	0.62	1.62	0.38	1.00	1.03	0.81	0.46	0.97	0.62	1.49	1.16	1.18	1.56	6.59	0.41	1.20	0.84	0.66	1.06	1.05	0.49	0.89	0.88	0.31	0.79	1.34	0.99	0.59
銀行	0.69	0.66	0.79	0.65	0.88	0.37	0.46	0.87	1.39	0.68	0.49	0.57	2.04	0.75	1.45	0.45	6.04	1.85	0.58	0.78	1.34	0.88	0.72	1.14	1.44	0.44	0.75	1.17	1.58	0.56
宏觀	0.39	0.33	0.27	0.33	0.57	0.15	0.30	0.76	0.67	0.30	0.39	0.18	1.27	0.48	1.42	0.30	0.54	8.65	0.21	0.39	1.66	0.57	0.12	1.15	1.24	0.15	0.33	1.57	0.94	0.30
企業	0.94	0.79	0.81	1.22	0.42	0.92	0.94	0.84	0.79	0.99	0.55	1.16	0.79	1.22	1.25	1.02	0.66	1.40	4.07	1.18	0.85	1.05	0.59	1.20	0.97	0.44	0.88	1.10	1.19	0.51
投資	1.06	0.70	0.81	1.04	0.42	0.60	0.92	0.83	1.01	0.89	0.55	0.76	1.22	0.91	1.42	0.75	0.75	1.85	1.01	4.13	1.09	0.88	0.61	1.45	1.12	0.40	0.86	1.37	1.36	0.55
通貨	0.49	0.75	0.49	1.01	0.30	0.20	0.17	1.41	0.54	0.34	0.57	0.34	1.81	0.80	2.01	0.34	0.49	1.80	0.20	0.32	7.71	0.97	0.15	0.89	1.34	0.39	0.37	1.35	1.39	0.64
同期	0.97	0.51	0.51	0.67	0.21	0.46	0.49	0.83	0.25	0.62	0.58	0.53	0.55	0.65	1.09	0.79	0.18	0.66	0.37	0.39	1.06	6.98	0.30	0.42	0.60	0.32	0.55	0.77	2.08	0.90
美元	0.87	1.14	1.11	1.13	0.74	0.83	0.69	0.93	0.82	1.72	0.73	0.76	1.15	0.78	1.12	0.93	0.92	0.99	0.76	0.98	1.07	1.03	4.75	1.03	1.03	0.53	0.90	1.03	1.29	0.80
資本	0.50	0.46	0.59	0.81	0.47	0.44	0.49	0.98	0.78	0.55	0.34	0.52	1.56	0.67	1.30	0.36	0.52	2.20	0.54	0.68	1.03	0.72	0.21	6.63	1.30	0.41	0.60	1.15	1.53	0.39
經濟	0.62	0.51	0.60	0.76	0.51	0.15	0.32	1.07	0.79	0.41	0.54	0.47	1.33	0.75	1.57	0.53	0.62	2.01	0.34	0.45	1.61	0.92	0.13	1.29	7.24	0.28	0.37	1.65	1.79	0.64
百分	0.29	0.29	0.42	0.21	0.13	0.13	0.17	0.92	0.13	0.21	0.63	0.17	0.08	0.25	0.38	0.34	0.08	0.25	0.17	0.34	0.50	0.54	0.34	0.34	0.34	4.38	0.25	0.48	0.25	0.17
市場	0.92	1.27	0.71	1.33	0.42	0.72	0.71	0.93	0.87	0.64	0.67	1.08	1.34	1.23	1.26	0.86	0.68	1.60	0.78	0.84	1.29	0.96	0.56	1.29	1.17	0.41	3.85	1.26	1.38	0.70
增長	0.98	0.38	0.44	0.70	0.32	0.28	0.32	0.76	0.28	0.60	0.47	0.32	1.33	0.76	2.34	0.35	0.19	1.36	0.22	0.19	1.26	0.60	0.03	0.76	1.01	0.25	0.38	7.74	1.33	0.57
季度	0.70	0.62	0.39	0.47	0.25	0.37	0.16	0.99	0.56	0.52	0.56	0.31	0.95	0.54	1.81	0.23	0.45	1.39	0.19	0.33	1.30	2.00	0.43	0.68	1.09	0.17	0.66	1.68	6.95	0.72
下降	0.79	0.75	0.67	0.90	0.41	0.45	0.32	1.38	0.37	0.58	1.03	0.41	1.04	0.91	1.79	0.55	0.36	1.30	0.31	0.41	1.54	1.83	0.49	0.95	1.15	0.40	0.63	1.49	1.93	4.91

Table A.17 Terms dependency in Domain Ontology Graph (軍事)

	武器	彈藥	軍事	裝備	導彈	戰爭	攻擊	裝甲	美軍	飛行	偵察	士兵	戰斗	雷達	飛行	空中	發射	艦船	陸軍	紅外	指揮	防務	坦克	國防	飛機	海軍	作戰	國防	空軍	部隊
武器	8.36	2.13	0.91	1.67	1.34	0.65	0.92	1.22	0.71	0.52	1.16	0.59	1.12	0.93	0.42	0.75	0.79	1.20	0.49	0.94	0.72	1.05	1.67	0.76	0.39	0.68	1.04	0.86	0.87	0.68
彈藥	1.02	10.86	0.45	1.17	0.68	0.23	0.61	0.90	0.42	0.49	0.49	0.42	0.76	0.04	0.34	0.49	0.34	0.25	0.45	0.23	0.76	0.30	1.29	0.42	0.26	0.26	0.79	0.34	0.61	0.68
軍事	1.00	0.81	4.65	1.24	1.11	0.70	0.81	1.23	0.96	0.49	1.23	0.97	1.08	0.87	0.39	0.80	0.82	1.03	0.95	0.51	0.96	1.29	1.43	1.07	0.48	0.99	1.21	0.98	1.00	0.90
裝備	0.99	0.96	0.48	7.56	1.09	0.27	0.59	1.50	0.51	0.76	1.22	0.43	1.35	0.92	0.62	0.83	0.76	1.05	0.55	1.11	0.54	0.99	1.64	0.59	0.51	1.09	1.35	0.64	1.06	0.53
導彈	0.98	0.53	0.75	1.34	10.05	0.48	0.80	0.91	0.59	0.81	1.25	0.25	0.88	2.61	0.42	0.82	1.93	0.96	0.38	1.30	0.48	0.88	1.28	0.65	0.66	0.96	0.86	0.67	1.15	0.39
戰爭	0.91	0.83	0.84	0.90	0.88	5.96	1.20	1.08	1.51	0.47	0.99	1.25	1.07	0.40	0.48	0.71	0.47	0.73	0.87	0.61	1.18	0.85	1.13	0.92	0.36	0.97	1.53	0.85	0.86	0.77
攻擊	0.78	1.00	0.64	0.92	1.04	0.74	7.77	1.19	0.77	0.54	1.06	0.64	1.20	0.92	0.60	0.81	0.96	0.93	0.50	1.08	0.72	0.68	1.44	0.50	0.41	0.72	1.12	0.51	1.02	0.73
裝甲	0.64	0.77	0.35	1.20	0.55	0.12	0.55	11.33	0.41	0.20	0.81	0.61	0.73	0.32	0.35	0.53	0.44	0.00	0.82	0.71	0.41	0.35	3.59	0.38	0.23	0.41	0.85	0.23	0.55	0.61
美軍	0.63	0.98	0.74	0.94	0.79	1.06	0.91	0.79	10.05	0.64	1.25	1.32	0.97	0.62	0.58	0.84	0.60	1.13	0.77	0.91	1.63	0.74	0.62	0.91	0.44	1.13	1.67	0.81	1.16	0.84
飛行	0.26	0.41	0.19	0.82	0.80	0.13	0.40	0.41	0.42	10.35	1.39	0.25	1.05	0.79	5.16	1.59	1.14	0.40	0.29	1.45	0.50	0.59	0.40	0.38	1.63	0.59	0.79	0.32	1.69	0.25
偵察	0.29	0.46	0.21	0.80	0.58	0.19	0.72	0.92	0.29	0.85	11.33	0.29	0.74	0.93	0.77	0.82	0.45	0.78	0.35	1.29	0.56	0.50	0.85	0.27	0.48	0.69	0.93	0.37	0.88	0.19
士兵	0.72	0.97	0.86	0.80	0.34	0.89	0.71	1.82	1.36	0.36	0.83	9.13	1.08	0.35	0.47	0.56	0.43	0.07	0.97	0.52	1.14	0.54	1.55	1.10	0.29	0.85	1.27	0.95	0.71	1.27
戰斗	0.76	0.75	0.58	1.41	0.80	0.44	0.88	1.23	0.67	1.06	1.29	0.64	8.76	0.76	1.02	0.98	0.53	0.70	0.51	0.87	0.74	0.76	1.52	0.65	0.69	0.98	1.24	0.65	1.44	0.68
雷達	0.47	0.06	0.51	0.92	2.27	0.21	0.57	0.41	0.33	0.88	1.19	0.18	0.80	11.33	0.90	0.61	1.33	0.77	0.20	2.86	0.31	0.57	0.55	0.61	0.72	0.59	0.72	0.63	0.90	0.16
飛行	0.20	0.29	0.20	0.50	0.48	0.15	0.33	0.35	0.52	4.04	0.73	0.22	1.01	0.52	10.79	1.33	0.31	0.07	0.17	0.80	0.35	0.37	0.22	0.26	1.44	0.39	0.66	0.26	1.44	0.24
空中	0.44	0.55	0.30	0.96	0.85	0.18	0.59	0.92	0.56	1.73	1.48	0.30	1.00	0.69	1.82	8.22	0.87	0.47	0.33	1.59	0.53	0.86	0.83	0.33	1.33	0.63	1.12	0.31	1.60	0.31
發射	0.53	0.63	0.54	0.98	2.06	0.19	0.77	1.10	0.46	1.30	1.43	0.27	0.59	1.61	0.42	0.95	10.23	0.65	0.41	2.19	0.45	0.68	1.76	0.52	0.80	0.72	0.64	0.47	0.98	0.40
艦船	0.26	0.25	0.00	1.30	0.78	0.26	0.78	0.00	0.43	0.35	0.83	0.17	0.78	0.69	0.09	0.52	0.52	11.33	0.00	1.05	0.35	0.35	0.52	0.17	0.52	2.42	1.04	0.35	0.52	0.35
陸軍	0.47	0.86	0.45	0.86	0.52	0.43	0.45	1.71	0.59	0.41	0.88	0.79	0.68	0.32	0.41	0.45	0.54	0.00	10.60	0.55	0.86	0.82	1.11	0.77	0.16	1.41	1.27	0.79	0.98	0.56
紅外	0.00	0.24	0.00	1.15	0.16	0.00	0.16	0.87	0.00	0.82	1.41	0.33	0.33	2.64	0.99	0.82	1.15	1.08	0.00	9.97	0.16	0.00	0.99	0.00	0.33	0.66	0.99	0.00	0.16	0.00
指揮	0.51	0.91	0.55	0.90	0.65	0.59	0.70	1.18	1.07	0.72	1.15	0.98	1.00	0.64	0.54	0.81	0.59	0.57	0.98	0.92	6.27	0.98	1.10	0.68	0.36	0.81	1.64	0.47	1.03	0.85
防務	0.42	0.37	0.35	1.07	0.69	0.21	0.32	0.55	0.28	0.60	0.77	0.28	0.79	0.60	0.42	0.93	0.58	0.68	0.49	0.14	0.69	9.65	0.65	0.88	0.32	0.95	1.16	0.72	0.83	0.42
坦克	0.77	1.19	0.28	1.12	0.96	0.14	0.44	4.14	0.30	0.21	0.92	0.19	0.77	0.47	0.26	0.49	0.68	0.46	0.40	0.85	0.21	0.58	10.44	0.26	0.35	0.58	0.70	0.19	0.49	0.28
國防	0.73	0.66	0.70	1.01	0.84	0.49	0.51	1.03	0.77	0.58	1.17	0.89	0.90	1.11	0.60	0.57	0.70	0.57	0.93	0.72	0.81	1.19	1.00	8.44	0.37	0.98	1.19	2.79	1.14	0.73
飛機	0.42	0.60	0.30	0.76	0.87	0.26	0.51	0.62	0.48	2.46	1.50	0.30	1.05	0.94	3.10	1.83	0.90	0.71	0.48	1.54	0.50	0.63	0.93	0.41	6.70	0.61	0.73	0.40	1.62	0.28
海軍	0.46	0.85	0.38	1.29	0.99	0.45	0.52	1.00	0.55	0.65	1.49	0.51	1.04	0.73	0.82	0.73	0.69	2.46	1.02	0.89	0.63	1.25	1.16	0.73	0.43	10.22	1.31	0.67	1.13	0.46
作戰	0.50	0.75	0.51	1.40	0.78	0.56	0.73	1.09	0.91	0.80	1.17	0.70	1.19	0.66	0.57	0.98	0.55	0.85	0.79	0.99	1.14	0.97	1.04	0.61	0.42	1.19	9.56	0.65	1.26	0.70
國防	0.81	0.67	0.70	1.11	0.90	0.56	0.56	1.02	0.78	0.60	1.12	0.82	0.95	1.05	0.60	0.60	0.78	0.68	0.89	0.65	0.76	1.38	1.08	2.96	0.38	0.98	1.15	7.74	1.15	0.73
空軍	0.42	0.62	0.37	1.09	1.09	0.28	0.71	0.81	0.59	1.63	1.42	0.45	1.37	0.74	1.71	1.35	0.83	0.40	0.72	0.97	0.72	0.78	0.91	0.78	1.00	1.07	1.19	0.60	9.25	0.40
部隊	0.79	1.24	0.90	1.16	0.60	0.64	0.87	1.76	1.00	0.46	1.15	1.32	1.11	0.40	0.42	0.76	0.67	0.59	1.08	0.57	1.22	0.87	1.54	0.93	0.33	0.88	1.42	0.90	0.96	7.37

Table A.18 Terms dependency in Domain Ontology Graph (醫療)

	療效	血壓	久遠	服用	中藥	踟躕	病情	戈壁	以免	血管	服藥	嘔吐	治療	患者	傳奇	止血	尋覓	荒涼	皮膚	臨床	血液	病人	傷口	注射	出血	醫院	藥物	疾病	疼痛	部位
療效	13.79	0.00	0.00	1.22	5.49	0.00	0.27	0.00	0.00	0.56	3.53	0.00	1.28	1.28	0.00	0.00	0.00	0.00	0.38	3.83	0.23	0.93	0.00	1.09	0.00	0.40	1.38	0.40	0.80	0.17
血壓	0.00	13.79	0.00	0.55	0.00	0.00	1.32	0.00	0.06	4.08	0.71	0.00	0.66	0.54	0.10	0.00	0.00	0.00	0.58	0.26	0.98	0.63	0.60	0.33	0.91	0.36	0.54	1.32	0.73	0.10
久遠	0.00	0.00	13.79	0.51	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.32	0.00	0.00	0.35	0.83	0.00	0.00	0.00	0.17	0.17	0.00	0.28
服用	1.45	0.50	0.21	13.06	1.53	0.00	0.61	0.00	0.11	0.78	3.93	1.29	0.41	0.77	0.24	0.81	0.00	0.00	0.21	0.83	0.90	0.49	0.27	0.91	0.84	0.19	2.34	0.49	1.23	0.09
中藥	7.99	0.00	0.00	1.87	12.17	0.00	0.42	0.00	0.00	0.00	0.00	0.00	0.00	0.91	0.00	0.00	0.00	0.00	0.59	3.93	2.13	0.00	0.00	0.00	0.00	0.00	1.21	1.21	0.00	0.00
踟躕	0.00	0.00	0.00	0.00	0.00	13.79	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
病情	0.40	1.49	0.00	0.75	0.42	0.00	11.82	0.35	0.09	0.69	1.64	0.63	0.72	0.95	0.16	0.90	0.55	0.00	0.33	1.25	0.36	1.09	0.15	0.42	0.99	0.35	0.70	0.57	1.18	0.16
戈壁	0.00	0.00	0.00	0.00	0.00	0.00	0.69	13.30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.38	0.00	0.00	0.00	0.50	0.00	0.00
以免	0.00	0.13	0.00	0.26	0.00	0.00	0.18	0.00	9.07	0.00	0.25	0.13	0.13	0.19	0.00	0.00	0.00	0.00	0.25	0.09	0.20	0.25	0.32	0.12	0.00	0.04	0.23	0.19	0.26	0.11
血管	0.74	4.10	0.00	0.87	0.00	0.00	0.62	0.00	0.00	11.91	0.00	0.00	0.62	0.62	0.39	0.00	0.00	0.00	0.76	0.73	1.70	0.53	0.00	0.93	0.43	0.17	0.56	1.23	0.91	0.29
服藥	4.06	0.63	0.00	3.81	0.00	0.00	1.28	0.00	0.10	0.00	13.79	1.21	0.51	0.51	0.18	0.00	0.00	0.00	0.20	0.89	0.48	1.08	0.00	0.57	0.39	0.21	1.54	0.31	0.84	0.00
嘔吐	0.00	0.00	0.00	1.57	0.00	0.00	0.61	0.00	0.06	0.00	1.51	12.48	0.51	1.40	0.00	0.00	0.00	0.00	0.12	1.64	0.45	0.60	0.32	0.53	0.48	0.38	1.02	0.25	2.84	0.11
治療	2.05	1.02	0.04	1.13	1.33	0.00	1.66	0.14	0.11	0.71	1.35	1.20	12.67	1.18	0.11	0.18	0.21	0.00	0.62	1.16	0.63	1.11	0.45	0.79	0.66	0.93	1.17	0.77	0.95	0.21
患者	2.16	0.79	0.00	1.06	0.96	0.00	1.46	0.00	0.14	0.69	1.23	1.47	0.47	11.31	0.04	0.00	0.00	0.00	0.40	2.23	0.66	1.40	0.26	0.62	0.63	0.32	1.17	0.71	0.88	0.16
傳奇	0.00	0.22	0.00	0.55	0.00	0.00	0.30	0.00	0.00	0.81	0.43	0.00	0.07	0.11	12.01	1.05	1.28	0.66	0.42	0.00	0.59	0.07	0.00	0.10	0.27	0.21	0.04	0.11	0.44	0.12
止血	0.00	0.00	0.00	1.13	0.00	0.00	1.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.63	13.79	0.00	0.00	0.00	0.00	3.42	0.77	5.49	0.00	2.80	0.73	0.00	0.00	0.00	0.00
尋覓	0.00	0.00	0.00	0.00	0.00	0.00	0.98	0.00	0.00	0.00	0.00	0.00	0.36	0.00	1.23	0.00	13.30	0.00	0.69	0.00	0.00	0.00	0.00	0.00	0.00	0.36	0.00	0.00	0.00	0.00
荒涼	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.77	0.00	0.00	13.30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
皮膚	0.71	0.83	0.20	0.33	0.75	0.00	0.41	0.00	0.16	1.08	0.32	0.16	0.49	0.46	0.28	0.00	0.48	0.00	11.57	0.82	0.70	0.51	0.95	0.52	0.10	0.22	0.49	0.76	0.77	0.46
臨床	4.39	0.23	0.00	0.80	3.09	0.00	0.97	0.00	0.04	0.63	0.88	1.31	0.44	1.22	0.00	0.00	0.00	0.00	0.50	11.82	0.69	0.89	0.00	1.23	0.85	0.26	1.45	0.52	0.75	0.19
血液	0.35	1.13	0.00	1.14	2.21	0.00	0.37	0.00	0.11	1.96	0.63	0.47	0.58	0.66	0.32	3.12	0.00	0.00	0.57	0.91	12.07	0.67	0.79	1.02	0.71	0.29	0.69	0.61	0.75	0.18
病人	1.36	0.70	0.17	0.60	0.00	0.00	1.08	0.00	0.13	0.59	1.37	0.61	0.44	1.18	0.04	0.68	0.00	0.00	0.40	1.14	0.64	11.65	0.29	0.92	0.75	0.37	0.85	0.74	0.93	0.20
傷口	0.00	0.84	0.52	0.43	0.00	0.00	0.19	0.00	0.21	0.00	0.00	0.41	0.35	0.35	0.00	6.10	0.00	0.00	0.94	0.00	0.97	0.36	11.72	0.38	2.11	0.48	0.28	0.55	1.69	0.35
注射	1.62	0.38	0.00	1.14	0.00	0.00	0.43	0.70	0.06	1.05	0.73	0.54	0.49	0.52	0.05	0.00	0.00	0.00	0.42	1.59	1.01	0.93	0.31	12.34	0.24	0.31	1.14	0.49	1.63	0.26
出血	0.00	1.08	0.00	1.09	0.00	0.00	1.04	0.00	0.00	0.50	0.53	0.52	0.71	0.75	0.15	2.60	0.00	0.00	0.09	1.14	0.72	0.79	1.77	0.24	12.41	0.31	0.40	0.49	1.44	0.30
醫院	0.87	1.18	0.09	0.67	0.84	0.00	1.41	0.27	0.16	0.72	1.00	1.13	1.68	1.09	0.18	0.53	0.22	0.33	0.59	1.01	0.59	1.34	0.81	0.80	0.83	9.99	0.69	0.50	1.37	0.25
藥物	2.09	0.58	0.08	3.10	1.17	0.00	0.72	0.00	0.13	0.60	2.14	1.06	0.56	0.93	0.04	0.31	0.00	0.00	0.37	1.91	0.67	0.86	0.26	1.23	0.40	0.17	12.32	0.54	0.43	0.11
疾病	1.04	1.40	0.18	1.03	1.10	0.00	1.05	0.18	0.16	1.53	0.85	0.84	0.61	1.37	0.14	0.00	0.00	0.00	0.66	1.47	0.69	1.12	0.36	0.77	0.58	0.24	1.13	8.22	0.94	0.27
疼痛	1.02	0.71	0.00	1.32	0.00	0.00	1.02	0.00	0.12	0.88	0.93	2.52	0.47	0.78	0.20	0.00	0.00	0.00	0.53	0.84	0.64	0.81	1.17	1.40	1.19	0.54	0.35	0.47	11.99	0.40
部位	0.43	0.20	0.24	0.20	0.00	0.00	0.27	0.00	0.10	0.55	0.00	0.19	0.10	0.29	0.11	0.00	0.00	0.00	0.63	0.42	0.30	0.34	0.49	0.45	0.50	0.03	0.20	0.16	0.79	9.34

Table A.19 Terms dependency in Domain Ontology Graph (電腦)

	計算	病毒	用戶	硬盤	連接	存儲	操作	硬件	內存	微軟	機器	版本	電腦	CPU	操作	代碼	廠商	兼容	NT	接口	IBM	應用	程序	軟件	系統	計算	編程	服務	數據	驅動
計算	5.90	0.20	0.48	2.27	0.18	0.90	0.43	0.70	0.14	0.80	0.43	0.41	0.62	5.11	1.18	0.87	0.35	0.29	0.09	0.73	1.75	0.63	0.22	1.14	0.26	5.11	2.55	2.03	0.53	0.25
病毒	0.29	12.55	0.35	0.63	0.11	0.57	0.16	0.22	0.16	0.20	0.15	0.15	0.43	0.00	0.58	1.10	0.42	0.22	0.07	0.81	0.00	0.23	0.20	0.42	0.20	0.81	2.11	1.79	0.34	0.00
用戶	0.50	0.17	11.87	1.21	0.30	0.83	0.17	0.53	0.15	1.46	0.37	0.24	0.70	0.00	2.23	1.33	0.35	0.00	0.09	0.00	0.93	0.75	0.26	1.21	0.24	0.70	3.40	3.11	0.37	0.35
硬盤	1.05	0.26	0.68	14.19	0.26	5.01	0.26	0.00	1.83	0.00	0.60	0.00	2.37	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.77	0.67	0.26	0.75	0.26	2.62	0.00	2.09	0.26	0.00
連接	0.22	0.09	0.43	0.61	8.87	0.28	0.26	0.32	0.46	0.38	0.50	0.10	0.48	0.00	0.56	0.54	0.35	0.00	0.11	0.00	0.31	0.37	0.18	0.44	0.37	0.33	1.38	1.05	0.33	0.24
存儲	0.51	0.26	0.67	7.12	0.17	14.19	0.68	0.84	1.78	0.00	0.29	0.19	1.20	0.00	1.10	3.13	0.00	0.00	0.00	0.00	2.44	0.77	0.60	0.85	0.34	1.27	2.67	2.03	0.34	0.23
操作	0.49	0.13	0.38	0.45	0.26	0.82	7.09	0.80	0.23	1.32	0.59	0.32	0.72	0.00	8.13	0.99	0.17	1.10	0.10	2.32	0.93	0.50	0.33	0.98	0.33	0.77	1.53	1.68	0.24	0.26
硬件	0.54	0.07	0.44	0.00	0.20	0.86	0.61	10.06	0.00	0.98	0.31	0.15	0.47	0.00	0.00	0.82	0.18	0.65	0.07	0.00	2.90	0.26	0.41	4.15	0.41	0.92	4.23	1.61	0.07	0.18
內存	0.16	0.00	0.21	4.41	0.48	3.01	0.32	0.00	13.70	0.00	0.00	0.70	0.63	0.00	0.00	1.94	0.00	0.00	0.00	0.00	0.00	0.41	0.16	0.45	0.16	0.20	0.00	0.00	0.16	0.00
微軟	0.41	0.11	1.10	0.00	0.18	0.00	0.74	0.90	0.00	12.65	0.21	0.33	0.77	0.00	9.45	1.35	0.59	1.42	0.07	1.32	3.69	0.24	0.33	2.42	0.11	1.01	8.06	4.39	0.07	0.10
機器	0.37	0.07	0.45	1.28	0.44	0.44	0.41	0.45	0.00	0.33	10.81	0.10	0.48	0.00	1.48	0.28	0.18	0.22	0.07	0.00	0.99	0.71	0.14	0.59	0.21	0.86	2.88	0.73	0.14	0.43
版本	0.48	0.14	0.31	0.00	0.10	0.31	0.34	0.24	0.67	0.56	0.11	13.33	0.39	0.00	2.48	0.00	0.26	1.39	0.10	1.73	0.00	0.12	0.19	0.62	0.24	0.48	1.51	1.53	0.10	0.52
電腦	0.66	0.33	0.98	4.07	0.38	1.54	0.63	0.78	0.34	1.28	0.55	0.27	5.21	0.00	3.44	2.23	0.39	0.59	0.07	1.74	2.61	0.48	0.29	1.71	0.30	1.33	2.29	3.10	0.30	0.20
CPU	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	14.19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
操作	0.25	0.17	0.86	0.00	0.17	0.53	2.08	0.00	0.00	4.84	0.47	0.74	1.08	0.00	13.78	3.04	0.88	4.80	0.00	5.94	1.19	0.32	0.25	2.61	0.17	0.93	2.60	5.94	0.00	0.00
代碼	0.57	0.42	0.92	0.00	0.28	2.68	0.57	0.69	0.98	1.23	0.16	0.00	1.13	0.00	5.44	13.51	0.37	0.00	0.14	5.05	0.00	0.18	0.57	1.21	0.14	1.23	4.42	7.86	0.14	0.00
廠商	0.12	0.12	0.47	0.00	0.30	0.00	0.12	0.29	0.00	1.05	0.20	0.27	0.36	0.00	3.08	0.73	11.68	0.00	0.06	2.14	0.86	0.46	0.18	0.51	0.00	0.45	1.88	0.95	0.06	0.32
兼容	0.30	0.00	0.00	0.00	0.00	0.00	0.90	0.73	0.00	1.74	0.17	0.99	0.75	0.00	11.50	0.00	0.00	12.94	0.45	0.00	2.14	0.77	0.00	1.07	0.45	0.56	0.00	4.75	0.15	0.00
NT	0.54	0.00	1.39	0.00	0.54	3.39	0.54	0.00	0.00	1.56	0.00	1.19	0.00	0.00	6.88	0.00	0.00	5.16	12.41	0.00	0.00	1.37	0.00	0.77	1.61	1.33	0.00	0.00	0.00	0.00
接口	0.52	0.00	0.00	0.00	0.00	0.00	1.04	0.00	0.00	1.51	0.00	1.15	0.52	0.00	13.29	6.32	1.37	0.00	0.00	12.65	0.00	0.66	0.00	0.74	0.00	1.29	0.00	8.23	0.00	0.00
IBM	0.63	0.00	0.61	4.36	0.00	1.99	0.63	2.31	0.00	3.20	0.53	0.00	0.63	0.00	2.01	0.00	0.42	1.51	0.16	0.00	13.64	0.80	0.00	1.12	0.00	2.14	4.91	4.99	0.47	0.42
應用	0.58	0.17	0.85	1.34	0.24	1.07	0.46	0.36	0.34	0.35	0.65	0.11	0.41	0.00	0.93	0.29	0.38	0.93	0.05	0.86	1.38	6.74	0.14	0.66	0.19	0.96	0.75	0.96	0.24	0.13
程序	0.44	0.31	0.49	0.87	0.22	0.99	0.75	1.08	0.44	1.05	0.59	0.35	0.58	0.00	2.01	1.53	0.62	0.00	0.13	0.56	0.22	0.48	8.46	1.05	0.39	0.94	3.93	1.25	0.19	0.25
軟件	0.68	0.18	0.91	1.00	0.20	0.79	0.54	3.78	0.25	2.40	0.36	0.36	0.91	0.00	5.05	1.31	0.28	0.86	0.04	0.64	1.28	0.44	0.27	10.98	0.27	1.42	3.92	3.41	0.32	0.14
系統	0.86	0.43	0.80	0.41	0.83	0.88	1.03	1.15	0.56	1.00	0.67	0.50	0.87	0.00	3.67	1.34	0.60	1.76	0.31	2.10	0.94	1.13	0.45	1.18	3.99	1.34	1.15	1.81	0.68	0.55
計算	2.87	0.27	0.50	3.29	0.15	1.13	0.41	0.80	0.10	0.95	0.50	0.26	0.74	0.00	1.71	1.26	0.24	0.43	0.06	1.06	2.33	0.61	0.22	1.35	0.28	7.19	3.70	2.82	0.31	0.16
編程	0.81	0.61	1.31	0.00	0.40	1.28	0.20	1.98	0.00	4.11	0.91	0.45	0.81	0.00	2.59	2.46	0.53	0.00	0.20	0.00	2.89	0.26	0.61	2.02	0.00	2.00	14.19	3.21	0.20	0.00
服務	0.92	0.46	1.34	1.59	0.29	1.09	0.52	0.84	0.00	2.50	0.26	0.51	1.15	0.00	6.62	4.90	0.30	2.21	0.00	4.10	3.28	0.37	0.40	1.97	0.11	1.71	3.59	13.76	0.52	0.15
數據	1.16	0.53	0.94	1.75	0.43	0.86	0.62	0.67	0.51	0.85	0.60	0.46	0.82	0.00	1.21	0.89	0.80	0.20	0.18	0.37	1.20	0.86	0.28	1.09	0.47	1.34	2.62	1.91	6.51	0.62
驅動	0.38	0.00	0.49	0.00	0.19	0.40	0.25	0.31	0.00	0.18	0.50	0.56	0.13	0.00	0.00	0.00	0.33	0.00	0.13	0.00	0.90	0.16	0.19	0.27	0.32	0.31	0.00	0.50	0.13	11.03

Table A.20 Terms dependency in Domain Ontology Graph (體育)

	決賽	男子	運動	選手	球賽	亞軍	對手	比賽	戰勝	動員	名將	亞運	世界	本屆	錦標	參加	球隊	運動	冠軍	奪得	亞運	擊敗	預賽	奧運	女子	金牌	教練	隊員	奧運	參賽
決賽	5.86	0.06	0.17	2.51	0.99	0.47	0.14	1.49	0.57	0.20	1.20	0.76	2.12	0.14	2.55	0.09	1.14	0.97	2.60	1.75	1.50	0.49	6.43	0.17	0.60	2.01	1.06	0.34	0.14	2.01
男子	1.48	5.62	0.30	1.31	0.66	0.16	0.19	1.02	0.20	0.34	1.07	0.73	0.43	0.08	1.67	0.25	1.18	1.09	1.02	1.04	1.64	0.27	1.56	0.12	2.08	1.72	0.92	0.33	0.14	0.67
運動	0.95	0.26	3.88	1.42	0.87	0.26	0.25	1.34	0.45	1.74	0.82	1.11	0.91	0.22	1.38	0.32	1.18	2.98	1.24	1.07	1.73	0.32	1.19	0.78	0.54	2.05	1.41	0.33	0.73	1.32
選手	2.85	0.09	0.22	6.02	0.18	0.18	0.17	1.67	0.21	0.57	1.39	1.26	0.51	0.13	2.26	0.08	0.67	1.45	1.84	1.57	2.49	0.24	4.34	0.36	0.84	2.48	1.24	0.25	0.32	3.05
球賽	1.75	0.26	0.39	0.28	5.83	0.00	0.07	1.84	0.39	0.33	0.00	0.00	8.82	0.13	2.35	0.13	3.50	0.46	0.84	0.13	0.00	0.53	0.00	0.39	0.26	0.12	0.57	0.20	0.39	0.90
亞軍	1.43	0.09	0.03	0.48	0.00	6.01	0.09	0.27	0.09	0.09	0.46	1.00	0.14	0.06	0.27	0.12	0.20	0.21	0.97	0.52	0.00	0.93	0.75	0.03	0.15	0.38	0.39	0.36	0.03	0.82
對手	1.16	0.12	0.15	0.99	0.28	0.27	4.59	0.77	1.44	0.31	1.33	0.49	0.54	0.38	0.78	0.28	0.68	0.50	1.06	1.47	1.53	2.71	0.36	0.28	0.51	0.87	0.67	0.32	0.29	0.36
比賽	2.81	0.08	0.25	2.28	1.62	0.22	0.15	4.86	0.35	0.64	1.01	0.82	1.67	0.13	2.20	0.14	1.87	1.62	1.88	1.32	1.62	0.32	3.95	0.47	0.67	2.06	1.45	0.32	0.43	2.59
戰勝	1.35	0.02	0.14	0.59	0.70	0.14	0.50	0.80	5.03	0.48	0.35	0.46	0.74	0.18	0.82	0.09	1.17	0.66	0.88	0.91	1.80	0.78	0.57	0.16	0.30	0.91	0.55	0.34	0.11	0.42
動員	1.01	0.07	0.52	1.43	0.51	0.15	0.07	1.32	0.39	4.50	0.86	1.06	0.66	0.19	1.41	0.13	1.13	4.17	1.22	1.08	1.97	0.17	1.46	0.65	0.48	2.45	1.45	0.23	0.59	1.55
名將	1.89	0.12	0.12	1.92	0.00	0.24	0.12	1.10	0.12	0.35	6.26	1.16	0.54	0.12	1.04	0.00	0.77	1.16	1.49	1.22	1.53	0.17	4.35	0.35	0.99	2.62	1.01	0.17	0.41	1.19
亞運	1.14	0.00	0.10	1.67	0.00	0.50	0.19	1.06	0.29	0.68	1.11	5.46	0.45	0.00	4.31	0.10	0.43	1.45	0.74	0.92	48.32	0.39	4.83	0.19	0.39	1.74	0.84	0.19	0.19	0.66
世界	2.83	0.18	0.24	0.60	6.66	0.06	0.18	1.55	0.42	0.60	0.46	0.40	6.26	0.06	2.66	0.06	2.79	0.84	1.38	0.57	1.57	0.12	0.00	0.78	0.36	0.65	1.04	0.42	0.66	0.54
本屆	1.09	0.04	0.21	1.17	0.34	0.11	0.26	0.90	0.37	0.35	0.74	0.47	0.33	4.62	0.79	0.49	0.51	1.48	1.04	1.31	0.46	0.48	0.88	0.28	0.44	2.19	0.80	0.14	0.28	1.40
錦標	2.41	0.33	0.41	1.88	1.25	0.08	0.16	1.22	0.24	0.73	0.62	2.72	1.89	0.00	6.17	0.08	0.72	1.14	2.30	1.87	8.58	0.08	0.00	0.90	0.57	1.77	0.89	0.08	0.90	1.30
參加	1.38	0.35	0.63	1.50	0.81	0.47	0.90	1.27	0.72	1.18	0.85	0.89	0.75	1.06	1.09	2.38	0.94	1.30	1.15	0.85	1.46	0.76	1.94	0.79	0.66	1.15	1.01	0.69	0.75	1.49
球隊	1.60	0.18	0.30	0.82	2.77	0.09	0.03	1.47	0.63	0.51	0.69	0.40	2.91	0.12	1.07	0.09	6.02	0.93	1.42	0.97	1.58	0.42	2.25	0.18	0.63	0.97	2.87	0.78	0.24	1.16
運動	1.15	0.08	0.51	1.60	0.56	0.10	0.05	1.52	0.29	2.52	0.96	1.23	0.72	0.16	1.65	0.11	1.27	5.99	1.41	1.19	2.30	0.12	1.70	0.69	0.52	2.82	1.67	0.20	0.62	1.72
冠軍	3.03	0.06	0.25	1.89	0.56	0.37	0.23	1.40	0.39	0.64	1.10	0.58	1.20	0.14	2.84	0.13	1.19	1.33	5.95	1.76	1.52	0.29	1.08	0.40	0.79	2.29	1.10	0.36	0.45	1.51
奪得	2.33	0.05	0.15	1.85	0.09	0.23	0.20	0.99	0.42	0.30	1.03	0.82	0.57	0.20	2.64	0.07	0.93	1.21	2.02	5.71	2.59	0.54	1.23	0.27	0.74	3.29	1.18	0.20	0.32	1.29
亞運	1.11	0.00	0.19	1.62	0.00	0.00	0.38	1.13	0.56	0.75	0.72	23.80	0.87	0.00	6.71	0.19	0.84	1.13	0.96	1.43	5.65	0.38	4.70	0.38	0.38	2.37	1.63	0.38	0.38	0.43
擊敗	1.38	0.07	0.18	0.54	0.77	0.93	1.16	0.60	0.69	0.22	0.42	0.59	0.31	0.33	0.20	0.13	0.94	0.27	0.77	1.48	1.17	5.79	0.56	0.09	0.29	0.44	0.34	0.29	0.11	0.56
預賽	5.51	0.00	0.21	3.28	0.00	0.21	0.00	1.86	0.21	0.41	2.37	2.76	0.00	0.00	0.00	0.00	1.38	1.04	0.80	0.79	5.45	0.21	6.26	0.41	0.83	2.62	0.90	0.21	0.21	3.29
奧運	1.15	0.04	0.21	1.52	0.62	0.13	0.12	1.42	0.40	1.23	0.84	1.00	0.93	0.21	1.57	0.10	0.75	2.51	1.24	1.15	1.88	0.18	1.95	5.95	0.50	2.70	1.13	0.21	2.79	1.37
女子	1.25	0.91	0.25	1.54	0.34	0.11	0.18	1.07	0.18	0.39	1.34	0.47	0.49	0.07	1.41	0.16	0.94	0.88	1.53	1.14	0.92	0.18	1.76	0.25	5.57	1.81	0.84	0.19	0.23	1.16
金牌	2.12	0.04	0.09	2.31	0.07	0.13	0.09	1.36	0.37	0.53	1.76	1.23	0.51	0.09	1.97	0.04	0.74	2.17	2.08	2.60	3.39	0.20	3.22	0.42	0.96	5.70	1.52	0.20	0.41	1.13
教練	1.43	0.14	0.23	1.47	0.44	0.17	0.14	1.42	0.31	0.82	0.87	0.76	1.05	0.06	1.27	0.09	2.77	1.73	1.27	1.19	2.98	0.20	1.42	0.37	0.57	1.94	5.47	0.43	0.20	1.10
隊員	0.93	0.20	0.34	0.55	0.61	0.39	0.25	0.74	0.34	0.41	0.43	0.45	1.15	0.09	0.60	0.25	2.50	0.47	0.78	0.34	1.19	0.29	0.56	0.18	0.27	0.53	1.37	5.42	0.20	0.82
奧運	1.08	0.05	0.22	1.56	0.62	0.16	0.12	1.40	0.39	1.26	0.80	1.00	0.86	0.21	1.50	0.12	0.77	2.46	1.27	1.16	1.89	0.17	1.94	2.99	0.49	2.69	1.12	0.23	5.96	1.40
參賽	2.29	0.17	0.32	3.06	0.58	0.31	0.05	1.72	0.10	0.82	0.86	0.50	0.46	0.32	1.56	0.20	0.94	1.55	1.47	1.10	0.66	0.12	4.37	0.50	0.67	1.22	0.92	0.32	0.52	5.02

List of English translation of the Chinese Terms

創作	creation
藝術	art
演出	perform
作品	works
觀眾	audience
藝術家	artist
文化	culture
演員	actor
劇團	troupe
節目	programme
音樂	music
歌舞	sing and dance
劇院	theatre
晚會	evening party
戲劇	drama
文化部	Ministry of Culture
舞蹈	dance
文藝	literature and art
舉辦	hold
表演	performance
舞台	arena
電影	movie
歌曲	song
戲曲	traditional opera
劇目	a list of plays or operas
精品	fine work
美術	painting
風格	manner
演唱	sing in a performance
展覽	exhibition
運輸	transport
鐵路	railway
公路	highway
總統	president
訪問	visit
主席	chairwoman
外交	diplomatic
會見	meet with
友好	friendly
外長	Foreign Minister
總理	prime minister
會談	talk
外交部	Ministry of Foreign Affairs
和平	mild
關系	relationship
議會	parliament
領導人	leader
今天	today
部長	head of a department
雙邊	bilateral
雙方	the two parties
表示	express
阿拉伯	Arabic
會晤	meet
邊關	frontier pass
抵達	arrive
大使	ambassador
委員長	head of committee
舉行	hold
巴勒斯坦	Palestine
共和國	republic
外交部長	Minister for Foreign Affairs
和平共處	peaceful coexistence
教師	teacher
學校	school
教學	teaching

車輛	cars
交通	traffic
公交	public traffic
旅客	passenger
列車	train
不忍	cannot bear to
客運	passenger transport
堅韌不拔	persistently
仁慈	kindly
客車	bus
交通部	Ministry of Communications
行駛	travel
運量	freight volume
公安	police
貨運	freight transport
鐵道	railway
公安部	Ministry of Public Security
星期二	Tuesday
駕駛員	driver
通車	be open to traffic
駕駛	drive
鐵道部	Ministry of Railway
公安廳	public security department
路局	railway bureau
違章	break rules and regulations
通行	have free passage
車站	station
污染	pollution
生態	ecology
環保	environmental protection
保護	protection
森林	forest
排放	discharge
污染物	pollutant
廢水	liquid waste
學生	student
辦學	run a school
中學	secondary school
培養	training
教育	education
素質	quality
小學	primary school
校長	principal
師資	persons qualifies to teach
校園	campus
高中	senior middle school
課程	course
畢業	graduate
教材	teaching material
家教	family education
家長	parent
學習	study
課堂	classroom
德育	moral education
大學	university
初中	junior middle school
老師	teacher
高等	higher
教委	State Education Commission
教職工	teaching and administrative staff
師生	teacher and student
學科	discipline
增長	growth
出口	export
企業	enterprise
收入	revenue
市場	marketplace
銀行	bank
財政	finance
美元	dollar

大氣	atmosphere	金融	finance
環境	environment	消費	consume
環保局	State Bureau of Environmental Protection	產品	Product
污染源	pollution source	投資	invest
自然	nature	下降	descent
野生	wild	百分之	per cent
污水	waste water	商品	goods
資源	resources	生產	manufacturing
垃圾	rubbish	同期	the corresponding period
水源	source of water	增長率	rate of increase
動物	animal	資本	capital
水污染	water pollution	經濟學	economics
野生動物	wild animals	貿易	trade
流域	valley	價格	price
人類	humanity	季度	quarterly
地球	the earth	幅度	range
水質	water quality	貨幣	currency
土壤	soil	增長速度	speed of increase
綠色	green	總額	sum total
回收	recovery	宏觀經濟	macro economy
防治	do prevention and cure	通貨	currency
植物	plant	大幅	large-scale
武器	weapon	治療	treatment
作戰	fight	病人	patient
戰斗	militant	藥物	medicines
美軍	U.S. Army	醫院	hospital
導彈	missile	患者	patient
海軍	navy	療效	curative effects
部隊	troop	踟躕	hesitate
飛行	flight	久遠	far back
艦船	naval vessel	尋覓	seek
國防	national defense	戈壁	desert
發射	shoot	荒涼	bleak and desolate
坦克	tank	傳奇	mythical
空軍	air force	皮膚	skin

雷達	radar
裝備	equipment
陸軍	army
軍事	military
偵察	reconnoiter
國防部	Ministry of National Defense
裝甲	armored
攻擊	attack
指揮	command
戰爭	war
士兵	privates
飛機	aircraft
彈藥	ammunition
空中	in the sky
防務	defense
紅外	infra-red
飛行員	pilot
軟件	software
用戶	user
程序	program
計算機	computer
硬盤	hard disk
操作系統	operating system
服務器	server
微軟	Microsoft
接口	interface
版本	edition
兼容	compatible
應用	application
計算	compute
CPU	CPU
內存	inner memory
硬件	hardware
操作	operation
NT	NT
IBM	IBM
血壓	blood pressure
血液	blood
疼痛	pain
嘔吐	vomit
服用	take
疾病	disease
血管	blood vessel
出血	bleed
以免	in order to avoid
病情	state of an illness
臨床	clinical
注射	injection
傷口	wound
服藥	take medicine
止血	stop bleeding
部位	position
中藥	traditional Chinese medicine
比賽	competition
冠軍	champion
選手	player
決賽	finals
女子	woman
運動員	sportsman
亞運會	Asian Games
亞運	Asian Games
金牌	gold medal
錦標賽	championship
隊員	team member
男子	man
奪得	compete for
運動	sports
動員	arouse
教練	coach
球隊	team
亞軍	runner-up
參賽	participate in a match

機器	machine	本屆	current
數據	data	戰勝	defeat
廠商	factories and stores	世界杯	World Cup
存儲	storage	奧運	Olympic Games
驅動	drive	球賽	match
病毒	virus	奧運會	Olympic Games
系統	system	名將	famous general
代碼	code	參加	join
編程	program	對手	competitor
連接	link	擊敗	beat
電腦	computer	預賽	trial match