AUTOMATED DATA DISCOVERY, REASONING AND RANKING IN SUPPORT OF BUILDING AN INTELLIGENT GEOSPATIAL SEARCH ENGINE

by

Wenwen Li
A Dissertation
Submitted to the
Graduate Faculty
of
George Mason University
in Partial Fulfillment of
The Requirements for the Degree
of
Doctor of Philosophy
Earth System and Geoinformation Science

Committee:

_____	Dr. Chaowei Yang, Dissertation Director
_____	Dr. Rob Raskin, Committee Member
_____	Dr. Ruixin Yang, Committee Member
_____	Dr. Paul Houser, Committee Member
_____	Dr. Peggy Agouris, Department Chairperson
_____	Dr. Richard Diecchio, Associate Dean for Academic and Student Affairs, College of Science
_____	Dr. Vikas Chandhoke, Dean, College of Science

Date: 07-27-2010

Summer 2010
George Mason University
Fairfax, VA

Automated Data Discovery, Reasoning and Ranking in Support of Building an Intelligent Geospatial Search Engine

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at George Mason University

By

Wenwen Li
Master of Science
Chinese Academy of Sciences, 2004
Bachelor of Science
Beijing Normal University, 2000

Director: Chaowei Yang, Professor
Department of Geography and Geoinformation Science

Summer 2010
George Mason University
Fairfax, VA

Dedication

I dedicate this dissertation to my parents.

Acknowledgement

First and foremost, I would like to express my deepest gratitude to my adviser, Prof. Chaowei Yang, for his guidance, encouragement and support in every stage of my graduate study. As my adviser, he tries his best to train me in many aspects to grow for my career. What he conveys to me is not merely the knowledge, but the confidence and strong mind to conquer any difficulty that comes into my life. His conscience, diligence and persistence in pursuing academic achievement inspires me. I am indebted to him more than he knows.

I am grateful in any possible way to my committee member, Dr. Rob Raskin for his supervision, advice and crucial contribution to my research and to this dissertation. Although Dr. Raskin is far way in California, he always takes time out of his busy schedules to meet and discuss with me. It was his encouragement and recognition that helped me to go through the hardest time when I was struggling about deciding on the topic of my dissertation. His involvement along with his originality and vision has triggered and nourished my research maturity. I will benefit from his counsel, for a long time in life.

I owe my sincere gratitude to my committee member, Prof. Ruixin Yang, who gives me important guidance and enormous help in my research, career planning and my personal life. Prof. Yang is the first person I want to approach when I meet with problems. He is always willing to help, always encouraging and always giving me great advice. I could not imagine how hard my Ph.D. life would have been without his guidance.

I would like to sincerely thank my committee member, Prof. Paul Houser, for his valuable advice and science discussion that have been of great value in this dissertation. When I narrowed my sights toward technical solutions, it was his broad vision that inspired me to look for creative solutions of problems from a higher level. I am proud that I had the opportunity to work with an exceptionally experienced scientist like him.

A special thank goes to my postdoc adviser, Prof. Michael Goodchild, for giving me the precious opportunity to work with him, and for releasing me from work to focus on my dissertation. His full understanding and intellectual support helped me to make this possible. I have the utmost respect for him, and feel very fortunate to have him as my future adviser.

I gratefully thank Mr. Doug Nebert from FGDC and Ms. Myra Bambacus from NASA for their continuous sponsorship in all the research projects I have attended during my Ph.D. study.

Collective and individual acknowledgments are also owed to the professors who have given me valuable comments and generous help in my research. They are Prof. Peter Fox and Dr. Ding Li from RPI, Prof. Donglian Sun and Prof. Liping Di from GMU, Prof. James Wilson from JMU, Dr. Bruce Wilson, Dr. Jerry Pan and Dr. Yaxing Wei from ORNL, Prof. Karl Benedict from UNM, Dr. Rahul Ramachandran from UAH, Prof. Naijun Zhou from UMD and Prof. Chuanrong Zhang from UConn. .

To my friends, Ms. Yang Tang, Dr. Weijie Zhang and Dr. Chunmao Ye, I would like to thank them for always being there for me. They are the best medicine that cure my homesick and loneliness. Without their care and love, I would not have been able to concentrate and finish my dissertation so soon.

To my friend Mr. Matt Carter, I would like to thank him for sharing his bright thoughts in Artificial Intelligence and machine learning with me, which were very fruitful for shaping up my ideas and research for this dissertation.

I thank all my colleagues Dr. Huayi Wu, Zhenlong Li, Kai Liu, Jing Li, Xin Qu, Qunying Huang, Min Sun and Rasaq Otunba, at the CISC, GMU for all the pleasant time we have spent in the past three years.

My special thanks goes to Mr. Steve McClure, Ms. Unche A. Saydahmat and Ms. Elizabeth A. Stuart for their great help in proofreading this dissertation and giving me valuable advice in improving my writing skills.

I would also like to thank all the volunteers who contribute their knowledge for the experiments in my dissertation.

Finally, I owe everything to my parents and my husband for their support.

Table of Content

List of Tables

List of Figures

List of Abbreviation

RSS	Really Simple Syndication
WWW	World Wide Web
XML	Extensible Markup Language
RDF	Resource Description Framework
RDFS	RDF Schema
OWL	Web Ontology Language
IR	Information Retrieval
AI	Artificial Intelligence
GUI	Graphic User Interface
KB	Knowledge Base
SWEET	Semantic Web for Earth and Environmental Terminology
PL	Propositional Logic
FOL	First Order Logic
DL	Description Logic
DLESE	Digital Library for Earth Science Education
VSTO	Virtual Solar Terrestrial Observatory
HTTP	HyperText Transfer Protocol
FTP	File Transfer Protocol
LSA	Latent Semantic Analysis
SVD	Singular Value Decomposition
OGC	Open Geospatial Consortium
FGDC	Federal Geographic Data Committee
ISO/TC211	International Organization for Standardization Technical Committee 211
WMS	Web Map Service
WFS	Web Feature Service
WCS	Web Coverage Service
CSW	Web Catalogue Service
NP	Nondeterministic Polynomial
EOSDIS	Earth Observing System Data and Information System
PNG	Portable Network Graphics
GIF	Graphics Interchange Format
GML	Geographic Markup Language
GWS	Geospatial Web Service

QoS	Quality of Service
ATF	Accumulative Term Frequency
SVN	Subversion
MFC	Microsoft Foundation Class
ODBC	Open Database Connectivity
RR	Refraction's Research
GIDB	Geospatial Information Database
WP	Webpage Priority
FIFO	First In First Out
NASA	National Aeronautics and Space Administration
NOAA	National Oceanic and Atmospheric Administration
SDI	Spatial Data infrastructure
ASDI	Arctic SDI
CUAHSI	Consortium of Universities for Advancement of Hydrologic Science
GCMD	Global Change Master Directory
GOS	Geospatial One Stop
NCDC	National Climatic Data Center
ECHO	Earth Observation ClearingHouse
COD	Collaborative Ontology Development
SPARQL	Protocol and RDF Query Language
MDSM	Matching Distance Similarity Measure
ANN	Artificial Neural Network
MLFFN	Multiple Layer Feed-Forward Neural Network
MSE	Mean Square Error
SMSE	Square root of MSE
ESIP	Earth Science Information Partnership
USGS	United States Geological Survey
$\mathcal{Q}$	Qualified Cardinality Restrictions
$\mathcal{N}$	Cardinality Restrictions
$\mathcal{O}$	Nominals
$\mathcal{H}$	Role Hierarchy
$\mathcal{S}$	An abbreviation for $\mathcal{ALC}$
$\mathcal{I}$	Inverse Properties

Abstract

AUTOMATED DATA DISCOVERY, REASONING AND RANKING IN SUPPORT OF BUILDING AN INTELLIGENT GEOSPTIAL SEARCH ENGINE

Wenwen Li, PhD

George Mason University, 2010

Dissertation Director: Chaowei Yang

In the field of geospatial data discovery, two goals must be met to bridge the gap between data providers and data consumers: (1) machine agent or a search engine must be able to identify the distributed data sources owned by data providers on the Internet, (2) the machine agent must also incorporate human intelligence to find the most suitable data sources required by data consumers.

To achieve the above goals, search algorithms are applied in the data discovery process so that a machine can implement automatic retrieval of needed information. However, most of the search algorithms focus on discovering general webpages rather than considering the characteristics of data sources in a specific domain, such as hydrology. This leads to the low performance of a search engine when handling domain-specific queries.

This dissertation presents a number of techniques that address the fundamental questions in the problem of geospatial data discovery: how to automatically discover and collect relevant geospatial data dispersed widely on the Web? Once this information is found, how can this information be encoded from human-readable format to machine understandable format? And how to make the machine incorporate human intelligence to answer various search questions?

This dissertation starts by developing an active crawler for automatic geospatial data discovery. Traditional data discovery methods include using general search engines, such as Google or accessing geospatial Web catalogues, such as Geospatial One Stop (GOS). However, Google aims to answer generic queries by treating all the keywords evenly without considering the special characteristics of geospatial data. If solely relying on Google, the needed services will be hidden in the long list of the search results. The drawback of using geospatial Web catalogues is that it assumes all data providers would register their services into the catalogues. However, this is apparently not true. In addition, the lack of timely updates generates considerable dead links in the catalogue. This dissertation proposes an accumulative term frequency based conditional probability model and develops a corresponding crawler to solve the above problem and discover geospatial data more efficiently.

This dissertation then examines the problem of building a domain Knowledge Base (KB) for modeling data and knowledge from multiple sources. Current approaches reported in the literature use a controlled vocabulary, which does not encode enough

logical relationships between spatial objects to enable semantic reasoning. To overcome this drawback, this dissertation proposes a new conceptual model to abstract, map, and model the geospatial knowledge for the hydrology domain. A Web-based tool is designed and developed for collaboratively populating the KB by users with different backgrounds according to the proposed conceptual model. In addition, a semantic reasoning procedure is implemented for locating all the suitable data candidates so as to enhance the performance of the geospatial search engine.

To provide the data consumers with the best resource, the search engine should be capable of automatically judging the similarities among spatial objects, like human beings do. Traditional statistical methods count the co-occurrences or shared information of objects to measure their similarity. However, human recognition of similarity is sometimes too complex to be simulated by simple mathematical equations. Given this reality, a neural network based feature matching model is proposed in this dissertation to realize an automatic similarity measurement based on the KB populated as suggested above.

Finally, this dissertation introduces two research projects: the USGS Arctic Spatial Data Infrastructure and the ESIP Semantic Web Testbed to demonstrate how the proposed methodologies are applied to domain applications to solve real-world problems.

Chapter 1: Introduction

The invention of the Internet has made a huge amount of information available for online sharing and browsing. However, much of this information is not well cataloged, and is too voluminous to be organized manually. Thus, there is a noticeable need for an automated tool that can effectively search the Web and locate needed information. The emergence of search engines has fulfilled this requirement. Presently, there are almost 800 million Internet users worldwide (Figure 1.1) and over 90% of these users visit websites after gathering information that is obtained from search engines [1]. A Nielsen report shows that there were 10 billion searches in June 2009 alone [2]. The field of search engines has, therefore, drawn a tremendous amount of research and attention in both academia and industry.

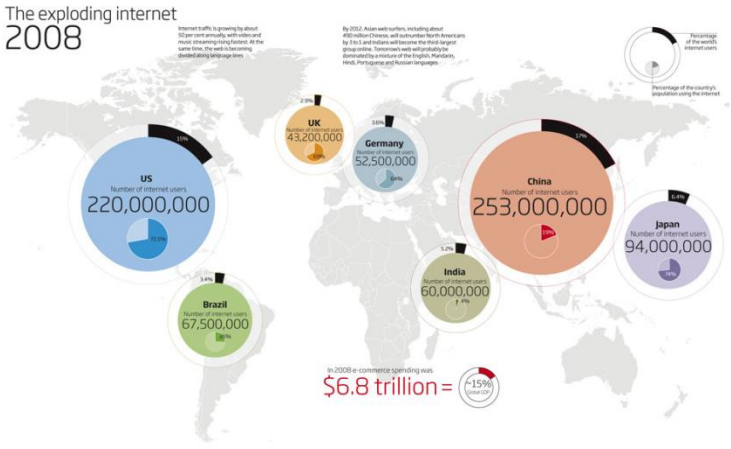


Figure 1.1 2008 Internet user distribution worldwide (Hand, 2009)

Among the huge amount of information available on the Internet, over 80% are associated with a spatial location on Earth. The linkage of information to the Earth gives general information extra value (BDO Consultants 1998); this type of information is called geospatial information or geographic information, which makes the spatial object easy to identify, analyze and reach. Because of this advantage, geographic information has a significant role in supporting applications in a variety of domains, such as water resource management [4], environmental modeling [5]-[6], navigation [7], transportation [8], urban planning [9], disaster management [10], and emergence response [11]. However, these disparate data are archived in various forms and are highly heterogeneous in data representation, storage and access [12]. Thus, solving the problem of semantic heterogeneity and enabling the efficient discovery of geospatial information becomes an essential task for search engines. This dissertation aims to study several key aspects towards building an intelligent geospatial search engine. These aspects include: (1) Data Collection by employing an active crawling technology to discover spatial information, data and services; (2) Data Searching by enabling semantic reasoning to capture latent logical relations between the users' query and candidate datasets and (3) Relevance Ranking by proposing an artificial neural network based feature matching model to improve the search performance.

1.1 Evolution of the traditional search engine

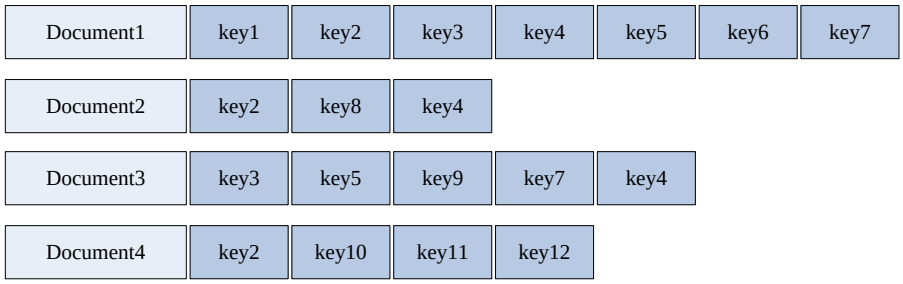
Yahoo! Search is the Web's oldest directory-based search engine, developed in 1994 by David Filo and Jerry Yang [13]. All of the webpages were reorganized into categories

and made searchable with a human compiled description for each URL. However, the exponential growth of the volume of pages on the Web posed a big challenge for Internet search engines to find needed information quickly and accurately. As the scalability of directory and human-directed search is limited, researchers [14]-[19] have turned to relying on new search technologies - Internet robots or crawlers for automatic webpage collection. WWW wander, developed by Gray in 1993 for measuring the growth of the Web, is considered to be the earliest Internet Web crawler. Another early attempt is WebCrawler, which was the first crawler to index entire webpages, and later became the original search technology for AOL (http://www.aol.com/). Google's Web crawling technology, proposed by Larry and Page in 1998 [19], has helped Google to receive more than 60% market share of searches (Figure 1.2) and as a result has been recognized as the most successful technology.

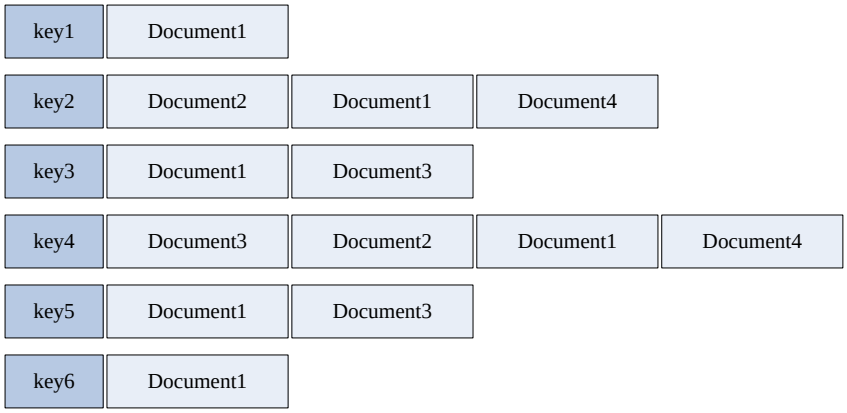
RANK	Provider	Searches (000)	Share of Searches
	Total	10,002,458	100.0%
1	Google Search	6,546,172	65.4%
2	Yahoo! Search	1,525,964	15.3%
3	MSN/Windows Live/Bing Search	1,073,416	10.7%
4	AOL Search	280,311	2.8%
5	Ask.com Search	177,589	1.8%
6	My Web Search Search	101,586	1.0%
7	Comcast Search	47,746	0.5%
8	NexTag Search	34,314	0.3%
9	BizRate Search	29,044	0.3%
10	Yellow Pages Search	25,260	0.3%
Source: The Nielsen Company			

Figure 1.2 Top 10 Search Providers for Nov. 2009, Ranked by Searches.

Google uses more than 100,000 parallel machines [20] to follow the hyperlinks within webpages, and at the same time, caches crawled web content. The cached web contents are stored in Google's index database for fast searching. The indexes are first built upon keywords extracted from cached web documents. The index table is represented as a Hashmap: the web documents are keys and extracted keywords are values. The advantage of this indexing method is that once a web document is cached, all its related keywords would be identifiable. However, this index cannot be used directly in the search interface because in a query, keywords are known while web documents are unknown. To solve this problem, an inverse index is built, where keywords are the keys and for each keyword, there is a list of web documents associated with it (Figure 1.3).



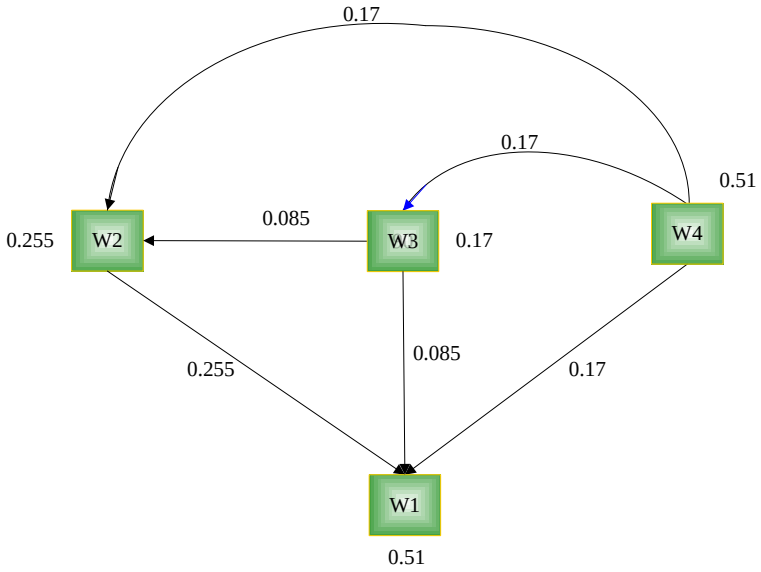
(a) Forward Index based on Documents.



(b) Inverted Index based on Keywords

Figure 1.3 Forward and Inverted Indices

This data structure allows rapid access to documents using query terms. In addition to the inverse index, Google employs a link analysis algorithm, Pagerank (PR), to sort the resulting web documents by their popularity to improve the search satisfiability. Pagerank is named after Larry Page, the inventor of Google. Its purpose is to measure the popularity of a web document within a set. The general idea of the PR algorithm is that a PR value of a webpage (W_i), represented by $Pr(W_i)$, equals to the sum of contributions from each of webpages (W_{i1}, W_{i2}…W_{in}) linking to it. If the webpage W_{ij} ($0<j<=n$) only has one outlink (W_i), the contribution W_i gets from W_{ij} ($0<j<=n$) equals to $Pr(W_{ij})$. If the webpage W_{ij} ($0<j<=n$) links to multiple webpages besides W_i, the contribution of W_{ij} to W_i equals to $Pr(W_{ij})$ divided by the number of outbound links $L(w_{ij})$. The below diagram demonstrates the PR distribution based on the given Web topology. Webpage W_1 has contributions from W_2 (PR (W_2)), W_3(1/2*PR(W_3)) and W_4 (1/3*PR(W_4)), so PR(W_1) equals the sum of the three scores (0.51).



The mathematical expression of the PR calculation is as follows:

where $\{W_{ij}\}$ is the set of webpages that link to $W_{i.}$ and d is the damping factor that represents the randomness of whether a user will continue to follow the links or restart from another random webpage. By default, d is set to 0.85 [19].

If the webpages $\{W_1, W_2 \ldots W_k\}$ have interlinks among them, the PR calculation can be represented as a linear function, as shown below.

Where $\ell(w_i, w_j)$ is 0 if there is not a link from w_i to w_j and the function is normalized such that $\sum_{i=1}^{k} \ell(w_i, w_j) = 1$[21].

From the above equation, the PR can be solved as:

$$\begin{bmatrix} PR(w_1) \\ PR(w_2) \\ \vdots \\ PR(w_k) \end{bmatrix} = \begin{bmatrix} \ell(w_1, w_1) & \ell(w_1, w_2) & \cdots & \ell(w_1, w_k) \\ \ell(w_2, w_1) & \cdots & & \vdots \\ \vdots & & \ell(w_i, w_j) & \cdots \\ \ell(w_k, w_1) & \cdots & \cdots & \ell(w_k, w_k) \end{bmatrix}$$

Because of the success of the above algorithms, Google became the dominant search engine on the market, receiving fully 63.98% of US searches by the end of June 2009, as reported by AVTEC [22].

Although the invention of search engines has facilitated the discovery of needed information for web surfers, current search engines still cannot fulfill users' requirements. These limitations can be largely grouped into two categories:

First, current mainstream search engines are based purely on the occurrence of words in documents without exploiting the hidden meanings. Thus, the precision of the results depends greatly on how users choose the query keywords. This makes many beginning Internet users feel discouraged and frustrated. Second, although search engines can return millions of webpages for a query, many of those are reported to be irrelevant. In addition, the behaviors of users in different communities are different; for example, Earth scientists desire to retrieve more Earth Science related information, while doctors and pharmacists

would pay more attention to webpages of their respective domains. Obviously, the above limitations of search engines are due to the lack of semantic-level support in terms of query understanding and domain-specific intelligence in web data.

Another limitation of mainstream search engines is their inability to provide instant information. The current trend of Internet query is trying to get "what's happening right now?" As mentioned by [23]:

"*40% of users perform search queries which display an intent that is best satisfied by realtime search results. Industry numbers aside, Iran – the country, the situation and the search query – has conclusively proven that users want search results from the realtime Web*"

But current popular search engines cannot meet this real-time requirement because queries are typically performed on a large-scale database index inside the search engines. Most of the search engines only update their index database on a monthly basis or even longer, so immediate news or newly launched webpages could not be found from these search engines. Even Google, which has the largest index database for search, conceded that it is not yet providing an appropriate search experience for such real-time information.

1.2 Evolution of real-time search engine

Filling this void produces a new concept called real-time search. Many new real-time search engines have been launched in the past few months. Popular real-time search engines include Scoopler, Collecta, Topsy, CrowdEye, TweetMeme, OneRiot, Yauba,

itpints, dailyRT and Almost.at. Among these, Scoopler (http://www.scoopler.com), launched in April 2009, aims to provide a single portal for users to perceive worldwide events instantaneously. It provides subject content in two ways: by tracking emerging content in real-time, and providing related information ranked by popularity. Collecta (http://collecta.com/) is claimed to be the fastest real-time search engine because of its adoption of the XMPP instant messaging protocol. It mainly collects and organizes web content from popular social media. Various search options are provided, such as blogs, comments, updates and video. Yauba (http://www.yauba.com/) is a cutting-edge real-time search engine study that is a joint effort of the Indian Institutes of Technology, the University of Delhi, MIT, Harvard University and UC Berkeley. It delivers multiple categories of information, such as traditional search, real-time search, mainstream search, video, images and even PowerPoint and word documents.

Topsy (http://topsy.com) is another real-time search engine and was launched by Topsy Labs, Inc. It aims to deliver search results as a real-time stream. Unlike other search engines, Topsy ranks content by the authority of the content's web sources and the number of times that others have shared specific content. When delivering results, it strikes a balance between the relevancy ranking and the immediacy of the information delivered. As more resources are considered in the ranking process, the information delivered becomes less real-time. Topsy allows rankings on hourly, daily, weekly and monthly basis, making it a good example to address the above trade-off.

Itpints (http://www.itpints.com/) is another attempt at weighing relevancy and immediacy, similar to Topsy. Unlike Yauba and Scoopler, itpints does not rely heavily on Twitter. Instead, it collects information from multiple online sources. Itpints can also deliver its information in the format of an RSS (Really Simple Syndication) feed, so it is not necessary for users to visit the website to view real-time information as it is created. Almost.at (http://almost.at/) has a distinct real-time search feature. The information it provides is based on 'event' rather than keyword search. By selecting a recent event from a list provided, almost.at is able to deliver related results. There are also many other real-time search engines such as CrowdEye (http://www.crowdeye.com/), TweetMeme (http://tweetmeme.com/) and OneRiot (http://www.oneriot.com/).

Although real-time search engines deliver instant updates and real-time flow of information and were believed to be the next big step in search engine functionality [24], they still have the following problems:

- Existing real-time search engines typically draw input data from social networking websites include Twitter (http://twitter.com/) and Digg (http://digg.com/), where anyone can publish anything. Thus, the reliability and authority of searched information are suspect.
- They lack support for specific domains, such as Earth Science.
- Search is still based on keyword matching, and lacks semantic query understanding.

1.3 Evolution of semantic search engine

Compared to the study of real-time search, the technology of which is more mature and industrialized, the study of semantic search is more theoretical. Actually, it has long been the dream of AI researchers to invent a machine that can understand the fundamental meaning of a human's request and provide an appropriate response. A search request is the most important type of request to understand and answer. This requires that both the request and web sources be well understood by the machine. However, almost all webpages on the Internet are written in plain text, without any meaningful tags or mark-up indicating what the web resources are or what they mean. This is easy for humans to digest, but is very hard for machines to interpret. To solve this problem, Tim Berners-Lee proposed the concept of Semantic Web in 1997 [25], aiming to augment the current World Wide Web (WWW) with a highly interconnected network of data that can be easily exploited and processed by both machines and human beings. Thus, the Semantic Web is designed to make web data more meaningful so that it can be understood, interpreted, manipulated, and integrated. To this end, W3C proposed a series of formal specifications to specify how web resources could be modeled, interpreted and presented. Some of these include Resource Description Framework (RDF) [26], RDF Schema (RDFS) [27] and Web Ontology Language (OWL) [28].

From this Semantic Web concept and traditional Information Retrieval (IR) technology [29], the state-of-the-art "Semantic Search" emerged [30]. The purpose of semantic search is to augment and improve the search process by leveraging XML and

RDF data from semantic networks to disambiguate semantic search queries and web text in order to increase relevancy of results [31]. It combines the research of traditional IR, Semantic Web, Artificial Intelligence (AI), and natural language processing, and has drawn a great deal of attention, from both academia and the industry, due to its potential to become a break-through technology in web search.

The Taalee Semantic Engine [32] was reported to be the first implementation of the semantic search idea. It uses intelligent agents to collect both a breadth and depth of information. "Breadth" is retrieved by crawling a wide range of web documents or from partners' resource repositories and "Depth" is realized by semantic asset cataloging, which is used to cross-reference and catalog the information into hierarchical categories correctly and automatically. The cataloged content forms a domain knowledge base, which is searchable from any site using semantic search facilities. This result has the potential to bring cataloging and searching to entirely new levels; however, technical details are sparse, and no fully implemented product is ready to use yet.

Guha et al. [33] proposed an idea of semantic search based on the semantic Web framework for "denotating" the search query to augment traditional search results with relevant data retrieved from distributed resources. The authors defined two types of search activities: navigational search and research search. Navigational search aims to search a phrase or combination of words that are expected in the documents. A typical navigational search is "W3C track 2pm Panel," this type of query does not denote any resource; instead users intend to get documents that contain these words. In contrast,

research search aims to get relative information about a concept, such as a musician or a geographical place. Sometimes, a simple navigational search on a keyword will cause ambiguity in understanding the category to which the concept belongs. In other cases, information from multiple documents may be needed to fulfill the user request. Semantic search focuses on this second type of search activities.

Guha et al. also discussed issues that should be addressed in a typical semantic search engine in order to augment the traditional search system. The first one is denotation, meaning that a semantic search system should provide a strategy to determine which node in the Semantic Web is the one intended by users and thus should be matched (called anchor node hereafter). The strategy could be the popularity of a term measured by its frequency of occurrence from a corpus or considering the context of users' search behavior. It could also be gleaned from users' profiles. The second issue is to determine "what to show," namely, what data should be incorporated in the search results. As the Semantic Web is an interconnected network, a large number of resources (triples) are connected with the anchor node. The authors proposed heuristic methods to include N closest nodes to the anchor nodes. Here the nodes could either be from the triples with the same source and same arc label, or having shorter distance to the anchor node. The third issue is how to format the results when presenting them to end users to guide navigation to get the information desired. The work by Guha et al. discussed the problems that a semantic search engine should solve and provided some feasible solutions. Meanwhile, it also left some open questions. For example, the methods they proposed to determine

relative resources to return to users had the potential to produce more unwanted information ("spam") and information that was irrelevant to the particular search context.

Rocha et al. [34] introduced a semantic search architecture that combines classical search techniques with spread activation techniques applied to a semantic model. Similar to the traditional search engines, this proposed approach provides a keyword-based GUI for query expression, and then the query is redirected to a full-text search system, Lucene [35]. Unlike in a traditional search, the full-text search does not match the query keywords with traditional web documents. Instead, it refers to matching the contents of the nodes in an instance graph created in the KB. A hybrid spread activation technique is employed to enhance the hit rate of the matching process by proposing cluster measurement (the percentage of concepts that one concept is related to, given that the other concept is also related) and specificity measurement (the inverse of the square root of the number of concepts of a given relationship that has a given concept as its destination) based weight mapping algorithm to assign a weight to each relationship by its importance. The cluster measurement tells how similar two concepts are and the specificity measurement tells how specific the destination concept is within a given relationship. Combining both measurement approaches, a combined measurement for weighing the importance of a relationship can be obtained. Based on the weight mapping techniques, the authors introduced a hybrid spread activation algorithm to obtain the closest set of nodes to a given query. Semantic inference occurs naturally in this process since both direct and indirect connections would be reasoned out from the initial set of nodes. The authors also define several constraints to control when to cease and where to

avoid the propagation, such as avoiding the nodes of a given concept type, ceasing the propagation at the nodes which are connected to more than a given number of other nodes, etc. Finally, the end results would be returned to the user. Note that because of the algorithm employed, documents that do not contain the occurrence of the initial query keyword could also be returned.

To test the architecture, the inventors chose a domain involving professor-student research information from their department's website and the Portinari project's website, which is an extensive KB regarding artwork. Experiments on the above domains showed that the proposed search engine produces satisfying results. This work is a direct extension of traditional search engines. Here the advantage is to employ the idea of weight mapping and spread activation to link concepts hidden in keyword descriptions from rich text attributes defined in semantic ontology. However, although this work poses some semantic advancement over traditional search, the inference done by spread activation still lacks semantic interpretation when the semantic network is propagated. Thus, it sometimes draws wrong inferences, and the quality of results is impacted.

The above semantic search studies have enhanced traditional search in some ways, but they still fail to meet expectations. The essential part of all semantic search engines is that they must rely on a well-defined KB in which facts (in triples) and rules are defined for inferring more facts and intelligent query understanding; however, it is extremely hard to have a KB covering all the facts in all aspects of life and to draw inferences based on it. Thus researchers in many fields have started to focus on building

KBs and further domain-specific semantic search engines. For example, in the Earth Science field, there are several well-known KBs, such as Semantic Web for Earth and Environmental Terminology (SWEET) [36] and semantic search engines, such as Noesis [37]. SWEET was the first practical KB (ontology) for Earth and environmental science. It categorizes terminologies into facets, including phenomena, property, substance and Earth realm to support reductionism. As the most popular ontology model in Earth Science, SWEET provides an upper-level abstracted expression of this domain. Founded on formal Description Logic (DL, [38]), it can support terminology reasoning (T-box reasoning). Based on SWEET, researchers from University of Alabama at Huntsville developed a semantic search tool to support an extended search vocabulary. The tool provides several inference capabilities such as equivalence, inversion and specialization for searching keywords. The web resources are from popular search engines such as Google and Yahoo! as well as educational databases such as the Digital Library for Earth Science Education (DLESE) and other scientific databases.

Another successful application is the Virtual Solar Terrestrial Observatory (VSTO) project, which provides a unified semantic portal to facilitate scientific data search from diverse data archives in the fields of solar, solar-terrestrial and space physics [39]. Unlike Noesis, which can be viewed as a semantic enabled recommendation system, VSTO is a heuristic and interactive data discovery system. Once a search candidate is confirmed in the current step, the system will move on to provide more specific options to close in on the exact needs of the end user.

1.4 Popular Search Methodologies

1.4.1 Faceted Search

Faceted search, also known as guided navigation, or parametric search, is a popular and intuitive interaction paradigm for search engines that allows users to digest, analyze and navigate through multidimensional data [40]. It is most commonly used in online shopping sites like Amazon (http://www.amazon.com), IBM (http://www-306.ibm.com/software/data/discovery), Endeca (http://endeca.com) and Mercado (http://www.mercado.com) to help users dynamically select a subset of interesting categories and drill down to matched subcategories. Typically, a facet search can answer the following type of questions:

When a user browses shoes matching the keyword "Boots," there will be a column displayed, together with the search results, to show the number of matches of the word "Boots" under each category, e.g., book, automotive, sports & outdoors, shoes, mp3 download, etc. If the category "shoes" is chosen, the results will be displayed and recommended items are then listed by brand, size, style, color and available number of products in each subcategory. The content of the navigation column is dynamically changed based upon the searching criteria. For example, if boots of a particular size are not in stock, that option would not be offered. Users can also choose to use a combination of criteria to filter from the complete result sets.

In comparison to traditional search like advanced form search or taxonomy search, multifaceted search has great advantages:

❶ Advantage over traditional advance search

Faceted search provides a more dynamic way to browse and search for resources than traditional "advanced search form," where all the available search fields are provided at once. In the traditional manner, users have to set up the search criteria at the very beginning of the search. However, users may not be completely clear about the keywords in all the dimensions when they initiate the search, so the traditional search is not suitable for typical searches. Moreover, enterprises try to populate additional valuable information to the existing structured template [41]. As a result, the structured properties may increase to a very large number, leading to a challenging search task and a loss of search focus. In addition, it is possible that users may want to select a combination of values that did not even exist in the document dataset [42], thus a navigational system provided by faceted search that can guide users to drill down to their areas of interest is an ideal solution.

❷ Advantage over taxonomy based search

Faceted search and taxonomy-based search have a lot of features in common. For example, they both organize the documents in categories and provide navigation based on the category classification. Therefore, Kehoe describes this as the next logical evolution of taxonomies [42]. However, faceted search is more powerful and dynamic than taxonomy-based search. Taxonomy tends to provide subject, product line-based search, in which the branches are mostly fixed [43]. In contrast, faceted search tends to be more dynamic; for example, in a geospatial search engine, a faceted search can accommodate multiple data types, such as dates, geographic coordinates, publication organization,

distribution source, etc. In mathematics, taxonomies are more tree-style, by which documents can only be navigated through pre-defined tree branches, whereas facets are graph-style, through which information can be viewed by the dynamic composition of multiple primary axes (Figure 1.4). In the future, the taxonomy-based search will be improved by introducing faceted search technologies to allow users have more granular control over the search results.

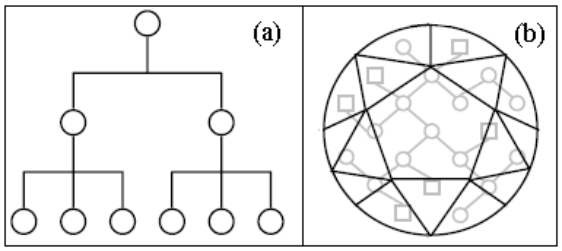


Figure 1.4 Taxonomy (a) v.s.. Facets (b) [44]

Ben-Yitzhak et al. (2008) described the basic design and runtime internals of a faceted search engine based on an open source search tool, Lucene. To be able to obtain the faceted navigation capability, several requirements should be met: (1) a faceted taxonomy built based on the content of the documents, (2) the rule to associate facets with each document in the collection, and (3) a strategy that can dynamically decide the facet path at run time.

The following example demonstrates of how faceted navigation will support the data discovery in a spatial search engine. Consider several datasets that cover the geographical regions of U.S., Canada, China, etc. The data are of either raster or vector format and are distributed through remote HTTP (HyperText Transfer Protocol) servers or FTP (File Transfer Protocol) servers. For example, dataset 1 covers the region of the U.S. and is

distributed by an HTTP server in raster format; dataset 2 covers regions of China and is distributed by a FTP server in vector format. Based on this information, a taxonomy index can be built (Figure 1.5).

The next step is to determine all the facet paths $P_1, P_2 ... P_{di}$ that are associated with each document d. Each $P_i = v_1 / v_2 / v_3 / .../ v_{k_i}$ denotes the whole path from root to leaf node in any tree of the document set in which this document exists. Meanwhile, each facet prefix ($v_1 / v_2, v_1 / v_2 / v_3 /, ..., v_1 / v_2 / v_3 / .../ v_{k_i}$) of the document is also recorded. A function $f(n)$ is defined to calculate the prefix once a document is linked to a node in the taxonomy index.

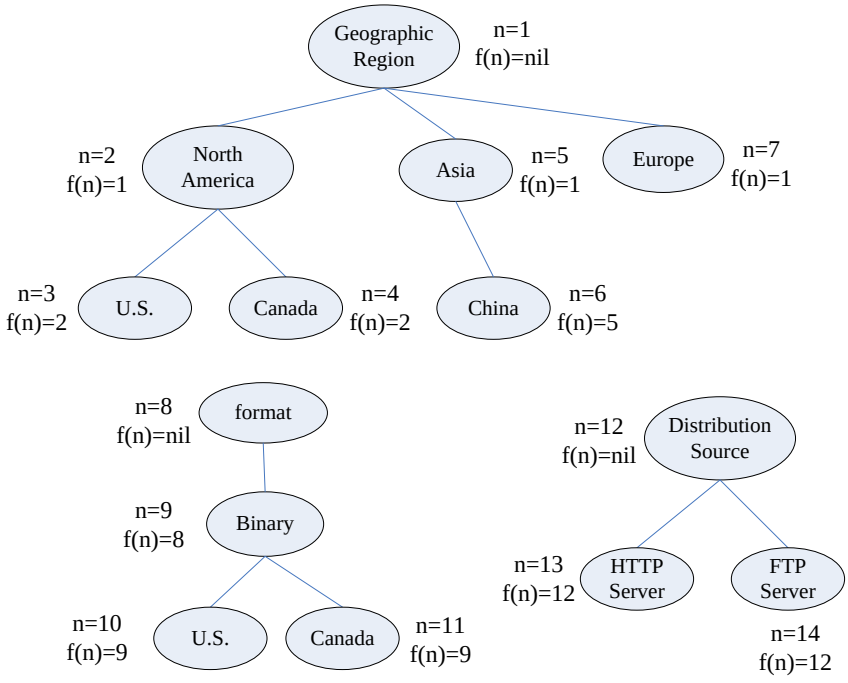


Figure 1.5 Taxonomy indexing.

Table 1.1. Inverted Index for Faceted Search.

facet$geographicRegion	Dataset1, Dataset2

facet$geographicRegion/NorthAmerica	Dataset1
facet$geographicRegion/NorthAmerica/US	Dataset1
facet$geographicRegion/Asia	Dataset2
facet$geographicRegion/Asia/China	Dataset2
facet$format	Dataset1, Dataset2
facet$format/Binary	Dataset1, Dataset2
facet$format/Binary/Raster	Dataset1
facet$format/Binary/Vector	Dataset2
facet$distributionServer	Dataset1, Dataset2
facet$distributionServer/HTTPserver	Dataset1
facet$distributionServer/FTPserver	Dataset2
Payload- Dataset1: 3, 10, 13; Dataset2: 6, 11, 14	

Table 1.1 shows the inverted index of the document to the prefix facets. Based on this inverted index, the payload of a document is recorded. Thus, a faceted search task can be performed as follows: (1) once a query is initiated, the Lucene engine will determine the documents that are related to the query; (2) for each document, find all the prefix facets based on the inverted index and the path determination function f ; (3) combine the prefix facets and generate a structured facet graph automatically (as Figure 1.6 shows); (3) count the number of documents that are within the scope known prefix facets; and (4) display the faceted navigation for refinement.

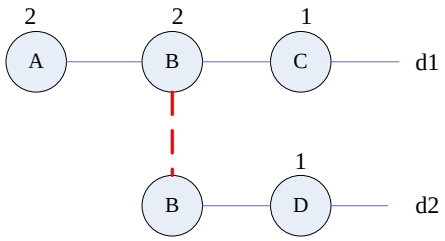


Figure 1.6 Facet Generation.

In this way, faceted search provides an interactive search paradigm for richer information discovery tasks over complex data models and becomes a popular search manner for current search engines.

1.4.2 Vector Space Search

Vector space approaches introduce linear algebra into automatic IR [45]. Data are modeled as a matrix; each document in the database is encoded as a vector, where the value of each member of the vector reflects the importance (or weight) of a specific term in representing the document. Then the search process identifies relevant documents through vector operations, such as to find the documents so that the distance between the document vectors and the query vector is smaller than a certain value. SMART [46] is one of the first systems to use the vector space model. Compared to other search algorithms, which need to preprocess (such as train) the datasets in advance to get good results, the vector space model is straightforward and usually more accurate. It has become the basis for many web search engines and is the most frequent approach used in commercial search engines.

Information Representation as a Vector Model

In a vector space model, all the unique terms that occur in all the documents in the data collection compose the vector. For each document, the value of each term is determined based on the importance of the term in the document. Typically, the value is larger if the term occurs more frequently in the document. A value is 0 means that the term does not occur in the current document. Suppose a document is described for

indexing and categorization purpose by a list of terms: **hydrological**, **search** and **engine**. It can then be represented by a vector in a three dimensional space. Figure 1.7 depicts an example when the terms have importance of 4, 3 and 0.45 separately. In this case, the word **search** is the most important one in the document, with **hydrological** and **engine** of secondary and tertiary importance.

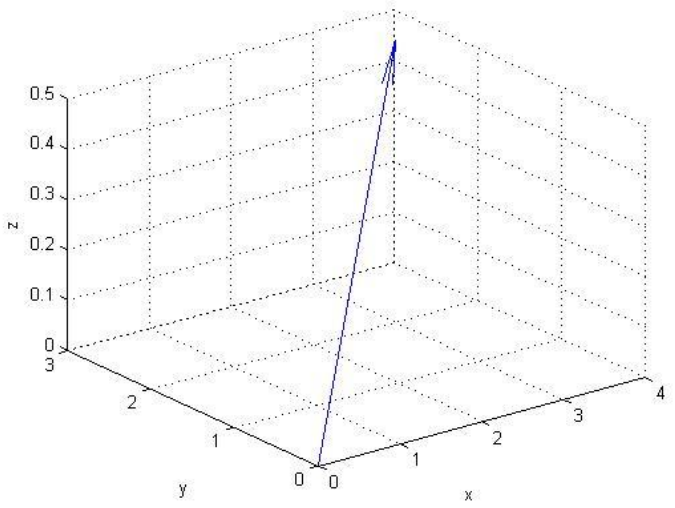


Figure 1.7 Vector Representation of a Web Document. X represents **search**, Y represents **hydrological** and Z represents **engine.**

A database containing d documents and t total terms comprises a term-by-document $t \times d$ matrix. Each matrix element, a_{ij}, is the weighted frequency of a term i to a document j. Figure 1.8(a) shows a small collection of titles where $d = 4$, and (b) is the list of all the validated terms ($t = 9$) with the stopping words removed. (Stopping words refer to the extremely common words, such as “the”, “an” and “any”). The 9*4 term-by-document

matrix A is illustrated in (c), in which a_{ij} is the number of times the term t_j appears in d_i. The column of A is the term vector and each row of A is the document vector. Figure 1.8 (d) shows A normalized for matching queries to each document vector.

Documents: d=4

D1: Darcy's law on hydrology (a)

D2: Law and Government for 142 Countries

D3: Fundamentals of Hydrology

D4: Hydrology and Water Law - Bridging the Gap

$$A = \begin{Bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \end{Bmatrix} \quad \text{(c)}$$

$A_N =$ (d)

$$\begin{Bmatrix} 0.5774 & 0.5774 & 0.5774 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5774 & 0 & 0.5774 & 0.5774 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.7072 & 0 & 0 & 0.7072 & 0 & 0 & 0 \\ 0 & 0.4472 & 0.4472 & 0 & 0 & 0 & 0.4472 & 0.4472 & 0.4472 \end{Bmatrix}$$

Terminologies: (b) t=9

T1: Darcy

T2: law

T3: hydrology(ic)

T4: govenment

T5: county(ies)

T6: fundamental

T7: water

T8: bridge(ing)

T9: gap

Figure 1.8 A Term-by-Document Matrix.

Query Matching Example

The process of query matching is achieved by calculating the similarity of a document to the specific query. In the vector space model, this problem turns into calculating the similarity of the document vector and the query vector. A simple visible measure of similarity is the cosine of the angle between the query vector and document vector. From the above figure, we can obtain that the document vector is a_i, where i in $Range\{rows(A)\}$. Assume a query vector is q, where $|q|=t$, the cosine of the two vectors can be computed according to the formula [50]:

$$\cos\theta = \frac{d_i^T q}{\|d_i\|\|q\|} = \frac{\sum_{j=1}^{t} d_{ij} q_j}{\sqrt{\sum_{j=1}^{t} d_{ij}^2}\sqrt{\sum_{j=1}^{t} q_j^2}}$$

for every document d_i in the document collection.

For example, suppose that a user in search of laws and rules on hydrology initiates a search for whitepapers about **hydrological law**. Thus, the query vector can be written as:

Un-Normalized: $q = [0\ 1\ 1\ 0\ 0\ 0\ 0\ 0]^T$

Normalized: $q = [0\ 0.7071\ 0.7071\ 0\ 0\ 0\ 0\ 0]^T$,

where the nonzero values are for the terms "hydrology(ical)" and "law." The document is considered as relevant only when the cosine value is greater than a threshold (in practice, the threshold is usually 0.9; we set it to 0.6 as the test document set is small). Calculating the cosines of angle of each document and the query vector, we obtain that

$$\cos\theta_1 = 0.8167$$
$$\cos\theta_2 = 0.4084$$
$$\cos\theta_3 = 0.5001$$
$$\cos\theta_4 = 0.6325$$

Therefore, the first document and the fourth document are returned as relevant documents, and the other two that rank low are correctly filtered out. However, the fourth document is a more satisfying result but it is ranked lower than the first document. Moreover, the basic vector space model which uses single word as indexing unit (as shown in Figure 1.8 (d)) may cause two different kinds of problems: (1) in the context of lexical atoms, the single words (such as "Darcy" in "Darcy's Law") cannot represent their actual meanings, and thus are very misleading if they are indexed separately; (2) the words may not be specific enough to carry the regular meaning, such as the "Law" in the phrase "Darcy's Law" [47]. To solve the above problems, the IR community has developed several auxiliary algorithms to extend the basic vector space model, such as using Latent Semantic Analysis (LSA, [48]) to provide a low-rank approximation to matrix A. Detailed discussion can be found in the next Section.

1.4.3 Latent Semantic Analysis

LSA is a variant of the vector space model that uses low-rank approximation to a vector space representation of the document set ([48][49]). LSA uses linear algebra theory to improve automatic IR, rather than relying on human-constructed vocabulary, KBs or semantic networks in traditional natural language processing or AI. The assumption of LSA is that there is some underlying or latent structure in the pattern of word usage across documents [50], and by uncovering this latent pattern, the dimension

of the term-by-document matrix A can be simplified in order to lower the computational complexity. The foundation of LSA is Singular Value Decomposition (SVD), by which the matrix A is decomposed into three matrices: (1) a term-by-concept matrix W describing the original column vector as an orthogonal unit vector; (2) a concept-by-document matrix P describing the original row vector as an orthogonal unit vector; (3) a diagonal matrix S containing the scale values. The mathematical expression is as follows:

$$A = WSP$$

Let's show an example and analyze how the technique works and what the LSA can accomplish. Suppose the data collection consists of titles of eight documents and two disjoint topics. As Figure 1.9 shows, c1-c4 is about geospatial semantic search and m1-m4 is about hydrological law. The dimensions of the term-by-document matrix A are 7×8 (Table 1.2). Seven is the number of terms that have occurred at least twice in the total documents and eight is the number of documents. Therefore, each row is the term vector and each column is the document vector. The SVD of the matrix A is shown

in Figure 1.10. The multiplication of X, S and P' perfectly constructs the original raw matrix A.

c1: The *geospatial* Web: how *geo*-browsers, social software and the Web 2.0
c2: *Geospatial semantics*: capture meanings of *spatial* information
c3: A *semantic search* engine for *spatial* Web portals
c4: Google's *spatial search* tools in the Marine *Environment* - Decision Support
m1: Darcy's *law* on *hydrology*
m2: *Hydrology* and Water *Law* - Bridging the Gap
m3: *Hydrology*: an *environmental* approach
m4: *Environmental law*: Hazardous wastes and substances

Figure 1.9 Example: A Collection of Eight Documents.

Table 1.2 The Term-by-Document Matrix A.

Matrix A	c1	c2	c3	c4	m1	m2	m3	m4
Geo	**1**	**1**	**0**	**0**	**0**	**0**	**0**	**0**
spatial	1	1	1	1	0	0	0	0
semantic	0	1	1	0	0	0	0	0
search	**0**	**0**	**1**	**1**	**0**	**0**	**0**	**0**
Environment(al)	0	0	0	1	0	0	1	1
law	**0**	**0**	**0**	**0**	**1**	**1**	**0**	**1**
hydrology	0	0	0	0	1	1	1	0

r(geo search) = -0.3333

r(geo law) = -0.4472

$A = XSP'$

X =

[	-0.3247	0.1477	0.4720	-0.5602	-0.0000	-0.0521	0.5774
	-0.7024	0.1891	0.0535	-0.0593	0.0000	-0.3622	-0.5774
	-0.3828	0.1566	0.2269	0.3804	-0.0000	0.7955	0.0000

-0.3777	0.0414	-0.4185	0.5010	0.0000	-0.3101	0.5774
-0.2914	-0.3849	-0.6299	-0.4960	-0.0000	0.3524	0.0000
-0.1163	-0.6197	0.2747	0.1430	-0.7071	-0.0807	-0.0000
-0.1163	-0.6197	0.2747	0.1430	0.7071	-0.0807	-0.0000]

S =

[2.7395	0	0	0	0	0	0	0
0	2.3709	0	0	0	0	0	0
0	0	1.6454	0	0	0	0	0
0	0	0	1.2380	0	0	0	0
0	0	0	0	1.0000	0	0	0
0	0	0	0	0	0.7963	0	0
0	0	0	0	0	0	0.0000	0]

P=[

-0.3749	0.1420	0.3194	-0.5004	0.0000	-0.5202	0.4649	-0.0062

-0.5147 0.2081 0.4573 -0.1932 -0.0000 0.4787 -0.4649 0.0062

-0.5340 0.1633 -0.0840 0.6641 0.0000 0.1547 0.4649 -0.0062

-0.5006 -0.0651 -0.6047 -0.0438 0.0000 -0.4017 -0.4649 0.0062

-0.0849 -0.5227 0.3339 0.2311 -0.0000 -0.2027 -0.1068 0.7086

-0.0849 -0.5227 0.3339 0.2311 0.0000 -0.2027 -0.1257 -0.7055

-0.1488 -0.4237 -0.2158 -0.2851 0.7071 0.3411 0.2325 -0.0031

-0.1488 -0.4237 -0.2158 -0.2851 -0.7071 0.3411 0.2325 -0.0031]

Figure 1.10 SVD Components of Matrix A.

Another approach to reconstruct the matrix A is to use only the most important two dimensions to implement *lower-rank approximation.* The approximated matrix is notated as $\hat{A}$. $\hat{A}$ (Table 1.3) is produced by multiplying part (as shown in the box) of each matrix. The lower-rank approximation collapses the component matrices in such a way that terms that occurred in the same contexts now appear with greater or less estimated frequency. Some terms that did not appear originally now do appear, at least fractionally [51]. For example, the term *hydrology* does not appear in document m4, therefore the corresponding cell A[7][8] is assigned 0. However, because m4 contains terms *environment* and *law*, the term *hydrology* is determined as related and its cell value has been replaced to 0.67 at $\hat{A}$ [7][8]. As a comparison, the term *environment* that appears in c4 and is assigned to 1 in A is now replaced by 0.4591 in $\hat{A}$ reflecting unimportance in characterizing the document based on context analysis.

Table 1.3 Two-dimensional Reconstruction of $\hat{A}$ from Original Matrix.

Matrix $\hat{A}$	C1	C2	C3	C4	M1	M2	M3	M4
Geo	**0.3833**	**0.5307**	**0.5322**	**0.4226**	**-0.1075**	**-0.1075**	**-0.0160**	**-0.0160**
spatial	0.7852	1.0837	1.1007	0.9342	-0.0709	-0.0709	0.0965	0.0965
semantic	0.4459	0.6170	0.6206	0.5008	-0.1051	-0.1051	-0.0013	-0.0013
search	**0.4019**	**0.5529**	**0.5685**	**0.5116**	**0.0366**	**0.0366**	**0.1124**	**0.1124**
Environment(al)	0.1697	0.2210	0.2773	0.4591	0.5449	0.5449	0.5055	0.5055
law	**-0.0892**	**-0.1417**	**-0.0697**	**0.2553**	**0.7951**	**0.7951**	**0.6700**	**0.6700**
hydrology	-0.0892	-0.1417	-0.0697	0.2553	0.7951	0.7951	0.6700	0.6700

r(geo search) = 0.9961

r(geo law) = -0.9655

The matrix reconstruction also brings changes between terms and multi-term documents. Considering the two pairs of terms {*semantic*, *search*} and {*semantic hydrology*} in the original matrix A and the reconstructed matrix $\hat{A}$, we evaluated the Spearman correlation coefficient r to measure the correlation between the terms in each pair. The mathematical expression is:

$$r(X,Y) = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}}$$

In the original matrix A, "geo" and "search," and "geo" and "law" never co-appear in any document. Thus, intuitively, they do not have much association. The Spearman correlation between "geo" and "search" is -0.3333 and the value is -0.4472 between "geo" and "law." However, in the reconstructed matrix, due to the uncovering of the hidden association, both values have greatly changed. The correlation between "geo" and

"search" increases to 0.9961 (almost the upper limit 1) and the correlation between "geo" and "law" decreases to -0.9655 (almost the lower limit -1). This is because both "geo" and "search" have high correlation with term "spatial" in the context, and the SVD analysis has uncovered this indirect relationship. Therefore, even though they do not co-occur in the same document, they occur in the contexts of similar meaning. For "geo" and "law," as no hidden association can be found in the context, they are ranked very low in the correlation analysis.

The above analysis demonstrates that LSA has significant advantages in uncovering the underlining relationships among documents. LSA can overcome the drawback of traditional IR in discovering documents on the same topic but using a different vocabulary. LSA can also improve the retrieval precision by distinguishing the terms with more than one meaning. However, LSA is less compact than the vector model and usually needs more storage space, which can impact the search efficiency during the IR process.

1.5 Problem statement and objectives

Per the aforementioned reviews of literature, we find that most of the above search engines, especially those for the Earth Sciences, are weak in understanding user behavior and providing the most relevant results. Success in search engines is not only a matter of quantity of the resources but also quality of the resources found. To be able to provide the exact information in a reasonable time, a search engine should incorporate the human intelligence to answer various questions. Therefore, the goal of this dissertation is to

study critical research issues in designing and implementing a semantic-enabled geospatial search engine.

Considering the shortcomings of existing systems, there are three critical problems to be solved:

- **How to enable the automatic discovery of geospatial data that exists in a distributed computing environment?**

  The richness and completeness of the backend data bank of a search engine determines its popularity. For example, Yahoo!, whose database is collected by humans, is readily surpassed by Google, which employs a machine robot to collect the Web data automatically. The Web has three characteristics that make automatic discovery (crawling) very difficult. First, the data on the Web is voluminous. Second, the status of the Web changes rapidly due to network, maintenance, and other reasons. Third, although the number of geospatial data and Web services is increasing dramatically, it is still a tiny small portion in comparison to the whole volume of the Web. Therefore, to locate them quickly and accurately on the Web to support decision making becomes a great challenge. A novel solution for the automatic discovery of geospatial data is discussed in Chapter 2, while the discovery of OGC (Open Geospatial Consortium) Web Map Services (WMSs) is taken as a case study.

- **How to conduct semantic related reasoning effectively?**

  The performance of a search engine relies heavily on how well it can understand the semantics of a user query as well as how fast it can provide a satisfactory answer.

This is the task of semantic reasoning. The basis for semantic reasoning is the KB from where the machine can infer new facts and knowledge. Therefore, defining a semantic schema and choosing a suitable formal language to represent the knowledge accordingly are essential to the success of a semantic reasoning engine. Propositional Logic (PL) can greatly reduce the computational complexity; however, it is not expressive enough for knowledge representation because the atomic unit for expressing knowledge is a statement, not individual words. In contrast, First-Order Logic (FOL) can express facts like natural language, but the reasoning task for FOL is a NP (Nondeterministic Polynomial) problem. How to implement an efficient reasoning service to balance between the expressiveness of a formal language and the computational complexity of reasoning is another question that this dissertation answers.

- **How to provide a proper ranking algorithm to improve the performance of the search engine?**

  Ranking helps a search engine to present the most relevant information at the top of the search results. Most search engine users only view the top 30-50 results, so even if the best answer exists in a result set, if it is not ranked highly by the tool, users will have a low chance of viewing it. In fact, ranking can be viewed as a similarity measurement process. All the available data resources can be considered objects with attributes and inter-relationships. Measuring how similar the objects are becomes a key issue. Usually, the objects are stored in a domain KB; when the size of the KB

increases, the difficulty of providing a complete similarity matrix (Time-Completeness Tradeoff) between each of the two terms becomes critical. In this dissertation, a machine learning algorithm-- a multi-layer feed forward neural network is proposed to learn and produce the similarity matrix automatically.

1.6 Summary of Contribution

The contributions of this dissertation are summarized as follows:

For the problem of automatic geospatial data discovery:

In Chapter 2, a novel algorithm is proposed based on conditional probability model and an associated Web crawler was developed to extend and improve the performance of existing crawling technologies in discovering the distributed geospatial data automatically. The purpose is to process Web URLs that are most likely to be WMSs earlier or to process webpages that are the most likely to contain WMSs faster.

The original contributions of this part of the work are:

- Significantly enhance the crawling effectiveness by proposing an accumulative conditional probability model and developing a crawler prototype.
- The automatic update mechanism keeps all the discovery geospatial WMSs up-to-date.
- Scientific discovery on distribution pattern of WMSs: dispersed globally and clustered locally.

Following this pattern, the OpenGIS community would be able to build a larger and more open environment for geospatial information sharing and interoperating. The above findings also provide effective guidance and principles for designing an efficient crawling engine to discover WMSs. Since the WMSs are Web services in nature, the algorithms used for discovering WMSs are also applicable for discovering other types of Web services. In addition, with the significant number of services identified by the proposed crawler, semantic relationships can be analyzed based on the content described in the capability file of each service and a KB can be built to allow semantic query from available services to enhance traditional search ability, as discussed in the following Chapter.

For the problem of knowledge-based reasoning:

In Chapter 3, a logical schema is proposed to abstract, map, and model the geospatial knowledge in the hydrology domain. A Web-based tool was also designed and developed for collaboratively populating the knowledge by users with different backgrounds. In addition, a semantic reasoning procedure was implemented for locating all the suitable data resources to enhance the performance of the geospatial search engine.

The primary contributions are:

- The proposed semantic schema that models hydrology knowledge enables intelligent reasoning for query expansion.
- The collaborative development tool reduces input specification requirements, therefore decreasing the registration burden by ontology contributors.

- The semantic reasoning model allows users to discover the right dataset needed without requiring expert-level knowledge.

The proposed semantic reasoning has inspired work in various directions. Different parts of the IR process can be augmented with semantic information. It is also becoming the core part for building a semantic search engine.

For the problem of semantic similarity determination:

In Chapter 4, a neural network based feature matching model is proposed to measure the similarity among spatial objects automatically. The collection and ontological modeling of spatial objects, the calculation of contribution for each feature of any two spatial objects and the neural network design are introduced. Compared to other existing methodologies, this approach has significant advantages in terms of:

- Providing a solution that can solve the large-scale similarity measurement problem automatically.
- The machine 'expert,' who gains the prior knowledge from training the neural network, provides an accurate similarity measures that can simulate human's recognition of similarity.

Given that semantic reasoning helps to find all suitable resources from the data repositories, semantic similarity measurement helps users to find the 'best' resource. By employing this technology, not only all related terms can be acquired, but also the relatedness can be quantified. Therefore, the recall rate of the geospatial search engine can be substantially improved.

The above discussion constitutes the theoretical contribution of this dissertation. Practically, this dissertation also verifies the feasibility of the proposed methodologies in support of real world applications.

Chapter 2 Data Discovery: An Active Crawler to Discover Distributed Web Resources

2.1 Introduction

The development of geospatial information acquisition methods helps to collect huge amounts of geospatial information. In 2006 alone, the NASA's Earth Observing System Data and Information System (EOSDIS) produced over 3 terabytes (TB) of Earth system science data on a daily basis [52]). As discussed in Chapter 1, the geospatial information is widely utilized in multiple applications. However, the data are archived in various forms, and the geospatial applications, provided by different vendors, are highly heterogeneous in data representation, storage, and access [12]. The heterogeneity makes it difficult to share and exchange geospatial information.

To solve this heterogeneity problem and to facilitate better sharing of geospatial information, standards have been adopted by a variety of organizations. In 1994 the Open Geospatial Consortium (OGC), the Federal Geographic Data Committee (FGDC), and the International Organization for Standardization/Technical Committee 211 (ISO/TC211) were established to develop a series of specifications and standards, such as FGDC Metadata Content Standards [53], WMS [54], Web Feature Service (WFS; [55]), and

Web Coverage Service (WCS; [56]). Most of the specifications leverage GWSs, also referred to as Distributed Geospatial Information Services (DGIS, [57]-[58]), to facilitate the sharing of geospatial information. The GWS defines software component interfaces that provide access to geospatial information through HTTP (Hypertext Transfer Protocol)-based queries.

For example, the WMS defines the interface for accessing geospatial data uniformly from remote servers in a standard format, such as PNG (Portable Network Graphics) and GIF (Graphics Interchange Format), through HTTP (Hypertext Transfer Protocol). Three WMS operations are defined and used in the following sequence: 1) "GetCapabilities" requests the service metadata; (2) "GetMap" requests a static map according to given geospatial and other parameters; and (3) "GetFeatureInfo" requests data of selected features. The three operations are issued to a WMS in the general format of http://WMS_URL? Request=Operations&Service=WMS&Version=1.*.* through the HTTP POST or GET protocols. The procedure allows WMS to integrate different geospatial information and services at the mapping level. Figure 2.1 shows a WMS link residing in a webpage as a hyperlink.

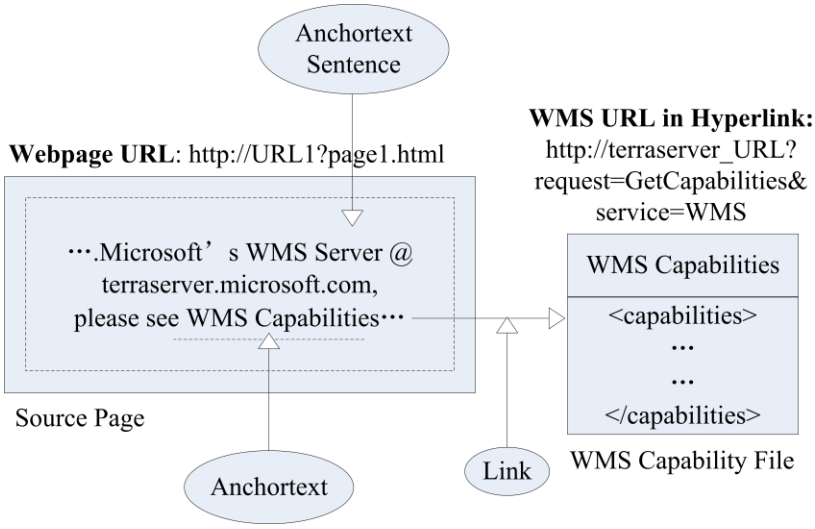


Figure 2.1. WMS’s Appearance on the Web: A web page may contain an anchor text with a Web link. The link may refer to a WMS URL for invoking a WMS capability file that describes basic service information for the WMS.

Other GWSs are defined to address interoperability at different levels, such as how WFS supports interoperation among vector data encapsulated by GML (Geographic Markup Language) and how WCS supports interoperation of geospatial data (such as GeoTIFF). The difference is that WMS provides an image map rather than the raw data [54]. WFS and WCS provide raw data to the client, as the data are generally geometric objects/features described by a set of vertexes and slightly primitive [55], which is not immediately visible compared to a static image returned by the WMS. Therefore, the implementation logic of WFS and WCS at the client side would be more complex.

The increased popularity of these standards has led to an increasing number of GWSs becoming publicly available on the Internet. Recently, researchers have explored automatic assembly of multiple GWSs to build distributed geospatial information systems

utilizing service chaining [59][60], Web service ontology modeling [61] and knowledge reasoning [62]. Besides, international organizations (e.g. OASIS and OGC) are discussing potential specifications, such as OASIS ebXML Registry Profile for Web Ontology Language [63] and OGC Catalogue Services - OWL Application Profile of CSW [64], to enable semantic description of Web service feature types, properties and science content in the registry to enhance keyword based search.

The prerequisite of all the above work is to have a significant number of live GWSs available [65]. Therefore, discovering the services in the open and dynamic environment of the Internet becomes a critical task [66][67]. Among several approaches for discovering GWSs, a centralized catalog with registered services is the most popular [68][69]. The catalog approach helps users to discover GWSs; however, it is based on the premise that service providers have registered their services in the catalog and the services are registered with correct classifications and with updated information. This assumption is frequently not met because many service providers do not register their services, and many service metadata entries in catalogs are not updated in a timely fashion [70][71].

The second approach is to utilize popular search engines, e.g. Google, to discover the services. But the popular search engines aim to answer generic queries by treating all the keywords evenly without considering the characteristics of GWSs. Although the number of existing GWSs is considerable, they are still a very small portion compared to the information volume on the Web. And the ranking mechanism of Google is based on the

number and weight of other links pointing to a certain webpage, which is not measured by the Quality Of the Services (QoS). So if only relying on Google to search, the needed service will be flooded and hidden in a long list of search results. Researchers have found that the Googled WMS, WFS and other accessible GWSs significantly under-represent the available OGC services [72]. Moreover, the GWSs always exist on webpages that are geospatial related, whereas Google's method is to crawl the entire web, which is not necessary here.

Observing the shortcomings of the two previous approaches, development of an efficient domain-specific crawling algorithm and the implementation of such a crawler becomes a compelling solution [73]. A domain specific crawler can improve search ability in both technical and economical perspectives. Technically, it is becoming increasingly difficult, if not impossible, for the general crawler to index the entire contents of the Web [74]. It is also an economic hurdle to build such a large-scale index database for a crawler. In addition, a domain-specific crawler can improve the quality of searches in terms of (1) allowing searches of pages that are currently not searchable from the general search engines, (2) providing a more up-to-date search, and (3) providing improved accuracy and extra features not possible with general search engines [75]. In the field of automatic WMS discovery, there is a so-called 'WMS-Crawler,' which crawls on the Web to find a hyperlink indicating a WMS and tries to parse it with a WMS Capabilities analyzer. This approach is promising; although unfortunately there are very few WMS-crawlers implemented and the performance of existing crawlers (RR[1], [67][76]) is not satisfying. This Chapter focuses on the research of the GWS crawling

algorithm [77] to: 1) address the shortcomings of both current catalogs and general search engines, 2) propose an ATF based conditional probability model to allow fast and automatic discovery and update of GWS on the Web, and 3) implement a high performance search engine which can be easily integrated into catalogs and a quality of service environment to support future automatic and smart service chaining. The discovery of WMSs is taken as an example.

2.2 Crawler Architecture

The crawler developed for WMS (2.2) includes Crawling Entry, Buffer, SourcePage Analyzer, Filter, Repository, R/R Handler, XML Resolver, and a database to archive the WMS and relevant metadata discovery.

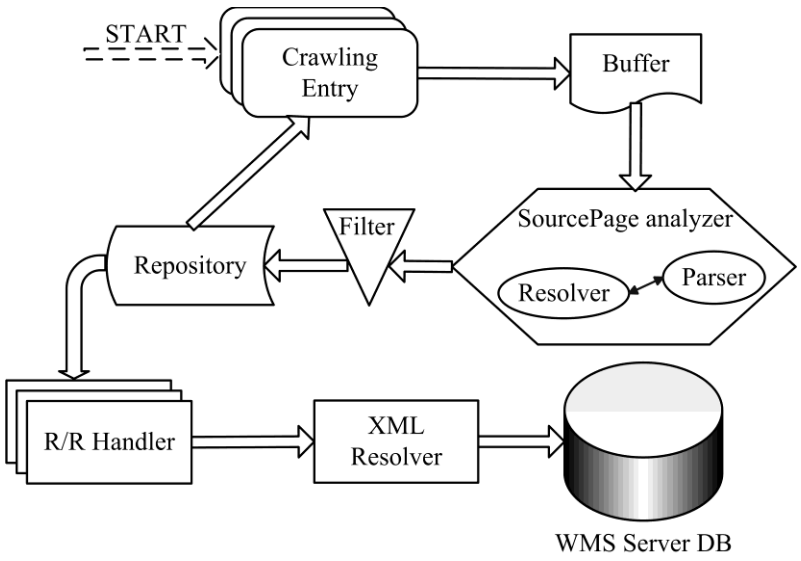


Figure 2.2 Crawler Architecture [78].

Crawling Entry is where the crawler starts to work. One or more seeding Web URLs are either identified by users or fetched automatically from the Repository. The

concurrent multi-threading technique detailed in Section 2.3.3 is used to crawl the Web for improving efficiency.

Buffer selectively caches web source code linked by URL. Different from other generic crawlers in caching all the Web documents they crawled for re-visit, our buffer only caches those having the possibility of containing a WMS URL by recognizing WMS characteristics: webpages not containing WMS URLs when crawled will have a low possibility of containing a WMS URL later on and are discarded. The hyperlinks cached in the buffer would be exported and stored to hard disk periodically. Once an unexpected error occurs, the crawling process can be recovered by loading the recorded information back to the memory. This buffering strategy helps reduce memory waste, lower the maintenance cost, as well as tolerate fault during crawling.

SourcePage Analyzer is used to analyze the web source code that has been cached in the buffer. It extracts all out-links and transforms the relative URLs of the links into absolute URLs. After completing the process, Analyzer removes the source code based on the strategy adopted for the buffer module. Then the URL goes to *Filter*, which utilizes priorities based on the probability of linking to a WMS before putting URLs into the repository. The Filter also filters out repeated URLs that already exist in the URL repository.

The *repository* maintains the URLs crawled using a priority queue, which is divided into several segments based on importance. *R/R (Request/Response) Handler* gets the URL at the head of the repository and sends the WMS GetCapabilities requests to the

server specified by the URL. The responses are parsed by the *XML Resolver* to validate the URL's status using the WMS tags, such as <WMS_Capabilities/>, <Service>, and <layer>. Once a WMS is found, metadata information such as layer numbers, service bounding box, spatial coverage, supported mapping projections as well as other QoS related information, including availability (whether a Web service is present or ready for immediate use), latency (the round trip time between sending a request and receiving the response) would be extracted and stored into the service catalog. The automatic metadata update will update the catalog periodically by a complementary model detailed in Section 2.3.4.

2.3. Techniques for improving WMS crawler

2.3.1 Prioritized Crawling: An ATF based Conditional Probability Model

It is very inefficient for a WMS crawler to visit all webpages on the Internet only to find a tiny part of needed resources. Therefore, quicker access to a webpage with a WMS URL will improve the performance. This efficiency can be achieved by assigning crawling priorities to webpages and URLs. Given a URL u, the probability of u referring to a WMS can be determined as follows.

Definition 1: Given a URL u, the probability of u is defined as $P(u) = (UP, WP)(u)$, where $UP(u)$ denotes to URL priority and $WP(u)$ denotes to priority of the webpage from which u is extracted, both $UP(u)$ and $WP(u)$ have the same range $\{0,1,2,3\}$.

The URL priority $UP(u)$ is determined according to the characteristic of a WMS. 1) The WMS is exposed in the format of a URL, which can handle a client's request through KVP (Keyword Value Pair) encoding, thus the URL must be linked to an active web page instead of a static webpage. Therefore, URLs linking to static webpages would not be considered as WMSs. But considering that the static webpages that the URLs link to might contain information related to a WMS, the crawler still reserve them but assign them the lowest priority $UP(u) = 3$. Some well-known image/document/video/audio formats can be excluded.

Higher priority would be given to the active pages. An Accumulated Term Frequency (ATF) based analysis is performed and the initial priority levels ($UP(u) = 0, 1 \, or \, 2$) are assigned based on the elementary statistical results. The statistical priorities change dynamically when more WMSs are identified. To obtain this information, the system maintains a list containing terms (atomic substrings extracted from URL string) in descending order of their frequencies. Initially, based on the first N (we chose N=50) WMSs found, we extracted the terms from their URL strings and calculated the TF values. Then the list was divided into three equal parts, the top one has the highest priority ($Pr(t) = 0$, t is any term in the top part), the middle one has the second highest priority ($Pr(t) = 1$) and the bottom one has the priority $Pr(t) = 2$. Each time a URL u is crawled, all the atomic substrings of the URL except protocol/hostname are extracted, represented by $T_u = \{t_1, t_2 \ldots t_n\}$. The priority $UP(u)$ is determined by the highest priority of term t_i in T_u , namely

$UP(u) = \min\{Pr(t_1), Pr(t_2) \ldots Pr(t_n)\}$ (Smaller value means higher priority). If there is no occurrence of t_i in the list, the priority is assigned as $Pr(t_i) = 2$. If this URL is later justified as a WMS URL, all terms will be extracted and both the term occurrences and frequencies will be updated. For example, suppose list L contains five terms in current stage, which is L = {"request=GetCapabilities"(t_1), 20; "service=WMS" (t_2), 16; "version=1.1.0" (t_3), 10; "cgi-bin" (t_4), 3; "servlet" (t_5), 3} in descending order of TF value. According to the section partition, the top two will be given the highest priority (0), namely $Pr(t_1) = Pr(t_2) = 0$, the following two have priority $Pr(t_3) = Pr(t_4) = 1$ and the last one has lowest priority among the current terms $Pr(t_5) = 2$. When a new URL u_x (http://hazards.fema.gov/wmsconnector/wmsconnector/Servlet/NFHL?request=GetCapabilities&service=WMS) is retrieved, the atomic substrings are extracted as T_{u_x} = {"wmsconnector," "wmsconnector," "Servlet," "request=GetCapabilities," "service=WMS," "NFHL"}. According to the definition of UP, we obtain that $UP(u_x) = \min\{\Pr(t_i), t_i \in L\} = \min\{2,2,2,1,0\} = 0$. As u_x is testified as a WMS URL later on, all the terms in T_{u_x} will be combined with those in L. The new L with updated TF is L ={"request=GetCapabilities" (t_1), 21; "service=WMS" (t_2), 17; "version=1.1.1" (t_3), 10; "servlet" (t_4), 4; "cgi-bin" (t_5), 3; "wmsconnector" (t_6), 2; "NFHL" (t_7), 1}. Partitioning L into three Sections, the priority for each term is obtained as: $Pr(t_1) = Pr(t_2) = Pr(t_3) = 0$, $Pr(t_4) = Pr(t_5) = 1$ and $Pr(t_6) = Pr(t_7) = 2$, which would

be used for future UP judgment. Through a continuously learning process, the size of L increases gradually to generate an enriched controlled vocabulary, which contains fundamental semantic information to guide future crawling jobs.

Based on Definition 1, the priority for URLs can be formalized as:

$$P(u_1 \mid UP(u_1) = i) > P(u_2 \mid UP(u_2) = j)\ ,\ where\ i,\ j \in \{0,1,2,3\}\ and\ i < j \qquad [1]$$

, which means that, if the URL Priority of u_1 is higher (smaller in value) than that of u_2, the probability that u_1 is a WMS URL is larger than the probability that u_2 is a WMS URL.

By investigating webpages that may contain a WMS URL, the probability of webpages containing keywords [Web Map Service] (we define $WP(u) = 0$) is higher than those containing [WMS, Service] (defining $WP(u) = 1$), which is higher than that of those containing [WMS, Server] (defining $WP(u) = 2$), which is higher than any others that do not contain these keywords (defining $WP(u) = 3$). To verify this hypothesis, the top 400 webpages returned from the Google search engine with the above query keywords were collected; tests show that 16% of the webpages that contain [Web Map Service] contain a WMS URL in the source code, and the ratio is 13.25% and 10% for those containing [WMS service] and [WMS server] respectively. Those pages without the keywords have a ratio of having a WMS URL near zero. So the priority of a webpage that may contain a WMS URL can be formalized as:

$$P(u_1 \mid WP(u_1)=i, UP(u_1)=k) > P(u_2 \mid WP(u_2)=j, UP(u_2)=k)\ ,\ where\ i,\ j,k \in \{0,1,2,3\}\ and\ i<j \quad [2]$$

, which means that when the UP of two URLs are equal, a lower WP implies a higher probability that it refers to a WMS.

2.3.2 Priority Queue

A new URL found from webpage out-links will be inserted into a priority queue (Figure 2.3) based on UP and WP defined by equations [1] and [2]. The URLs in the front of the queue have higher priorities and the queue is dynamic when inserting new URLs: UP decides which section the newly crawled URL belongs to and WP decides the position of a URL in a section. For URLs with the same priority, the latest crawled URL will be inserted behind those crawled previously. By this strategy, the crawling path is determined and recorded dynamically into the priority queue.

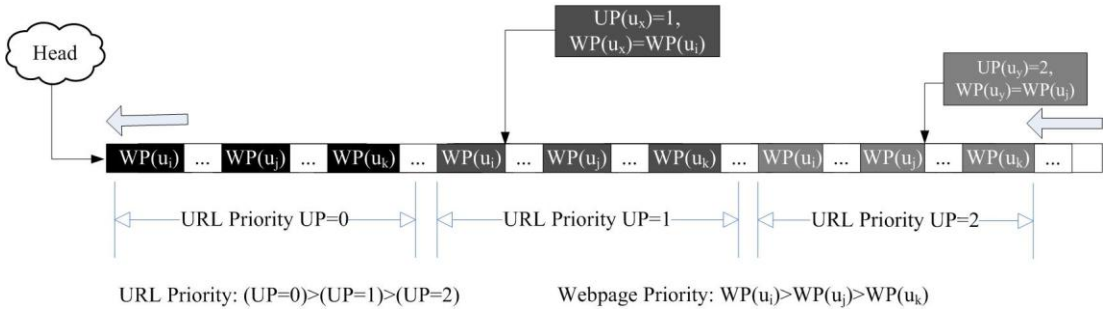


Figure 2.3 Data Structure of the Priority Queue.

2.3.3 Multi-thread

Crawling webpages to find URLs and analyzing webpages to find WMSs are the most time consuming components of the crawler. The processes are independent from each

other except when they try to access the same section of the queue. Therefore, two groups of multi-threads can be utilized to speed up the crawling and analysis respectively: 1) Group one crawls the WWW, extracts URLs, and inserts the extracted URLs into a priority queue. 2) Group two picks URLs from the priority queue and sends GetCapabilities requests to determine whether the URLs refer to live WMSs and updates the catalog. The priority queue is set as a critical section for the two thread groups. The concurrent threading could speed up the crawling but there are an optimal number of threads for performance tested in Section 2.5.1.

2.3.4 Automatic Update

The automatic update module (Figure 2.4) is adopted to keep the WMS metadata up-to-date in the catalog and resolve the metadata for service and layers. It works in two modes: 1) Message driven: when a WMS is retrieved by the crawler, the automatic update module will be invoked to resolve metadata; 2) Re-visit control: periodically, the module will pull the records out from the catalog and communicate with WMS servers to get the updated metadata. If the server is unavailable, the module will record it as an error and update the 'reason of unavailability' column in the catalog. The module could be added to *scheduled tasks*, so the operating system can invoke the module periodically to revisit and update WMSs' status. The revisit period can be of any length, depending on the liveliness requirement of various application systems. The WMS typically do not change as frequently as news websites do. Therefore, the update can be conducted as infrequently as once per day. Meanwhile, separating this module from the crawler

architecture helps to isolate this time-consuming process from the crawling tasks, so the automatic status update can be conducted in parallel with the crawling process. In this way, crawling efficiency is improved and the design complexity is reduced.

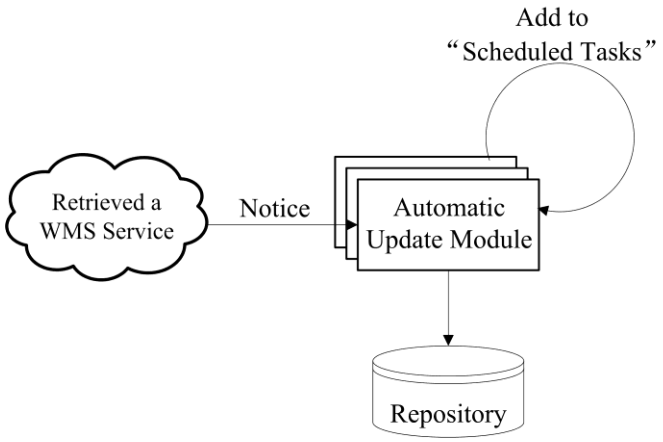


Figure 2.4 The WMS Metadata Automatic Update Module.

2.3.5 Algorithm Description

Figure 2.5 depicts the crawler's main procedure that manages two thread groups: CrawlWebPage and CheckWMSSite. Data structure URL_queue is the priority queue. The crawler's URL priority determination is implemented by function Priority() in Figure 2.6. Resolving the XML files returned from WMS is implemented in function Resolve() in Figure 2.7. The two thread groups maintain two cursors to the priority queue, one for crawling the Web, and the other for checking WMS URLs. Any group that reaches the end of the URL_queue will be suspended until a new URL is inserted.

```
Algorithm 1: Main Procedure
BEGIN
    EnterQueue (URL_queue, initial_URL, (0, 0))
    FOR i=0 TO (Max_Page_Threads-1) STEP 1 DO
        BeginThread (CrawlWebPage, thread_priority)
    END FOR
    FOR i=0 TO (Max_WMS_Threads -1) STEP 1 DO
        BeginThread (CheckWMSSite, thread_priority)
    END FOR
END
```

Figure 2.5. Pseudocode of Main Procedure.

```
Algorithm 2: ThreadGroup1

FUNCTION CrawlWebPage
BEGIN
    WHILE NOT all_been_visited (URL_queue)
    BEGIN
        URL = GetHeadCursor_Web (URL_queue)
        Webpage = crawl_page (URL);
        MarkAsCrawledPage (URL_queue, URL)
        url_list = extract_urls (Webpage)
        FOR EACH u IN url_list
        IF u NOT IN URL_queue THEN
            (p0_u, p1_u) =Priority (u)
            EnterQueue (URL_queue, initial_URL, (p0_u, p1_u)))
        END IF
        END FOR
    END WHILE
END FUNCATION
```

Figure 2.6. Pseudocode for Crawler ThreadGroup1.

```
Algorithm 3: ThreadGroup2

FUNCTION CheckWMSSite
BEGIN
    WHILE NOT all_been_visited_for_WMS (URL_queue)
    BEGIN
        URL = GetHeadCursor_WMS (URL_queue)
        ReturnedFile=SendGetCapabilitiesRequest(URL);
        IF ReturnedFile CONTAINS WMS_Metadata_Tags THEN
            PutInDatabase (Resolve (ReturnedFile))
        END IF
    END WHILE
END FUNCATION
```

Figure 2.7 Pseudocode for Crawler ThreadGroup2.

2.4. Prototype Implementation

The proposed techniques were implemented to improve efficiency into a prototype based on Microsoft .net framework. The main goal of the prototype is to enable an end user to customize the process of WMS crawling and to monitor system resource usage during crawling. The prototype is open-source and accessible through a SVN (SubVersion) server. Major technical details are as follows:

- Crawler is coded using Visual C++, and the Graphic User Interface (GUI) of the crawler is created using MFC (Microsoft Foundation Class)
- Automatic update model is coded using Visual Basic Script and is made executable using Windows Task Scheduler
- Backend database adopts Microsoft SQL Server 2005 SP1, and its connection with the crawler is through Microsoft's ODBC (Open Database Connectivity) driver.

Figure 2.8 shows the GUI designed for a general user to utilize the prototype of the proposed crawler. The top to bottom of the left side of the GUI is designated as the start point of the crawler; crawler status reports, including start time and duration of the crawling task; the number of crawling threads of the two groups; search results, including webpages that have been crawled and the WMS address if any WMS is found. The right side of the GUI shows statistical information including average processing speed and hardware usage for monitoring the crawler's status.

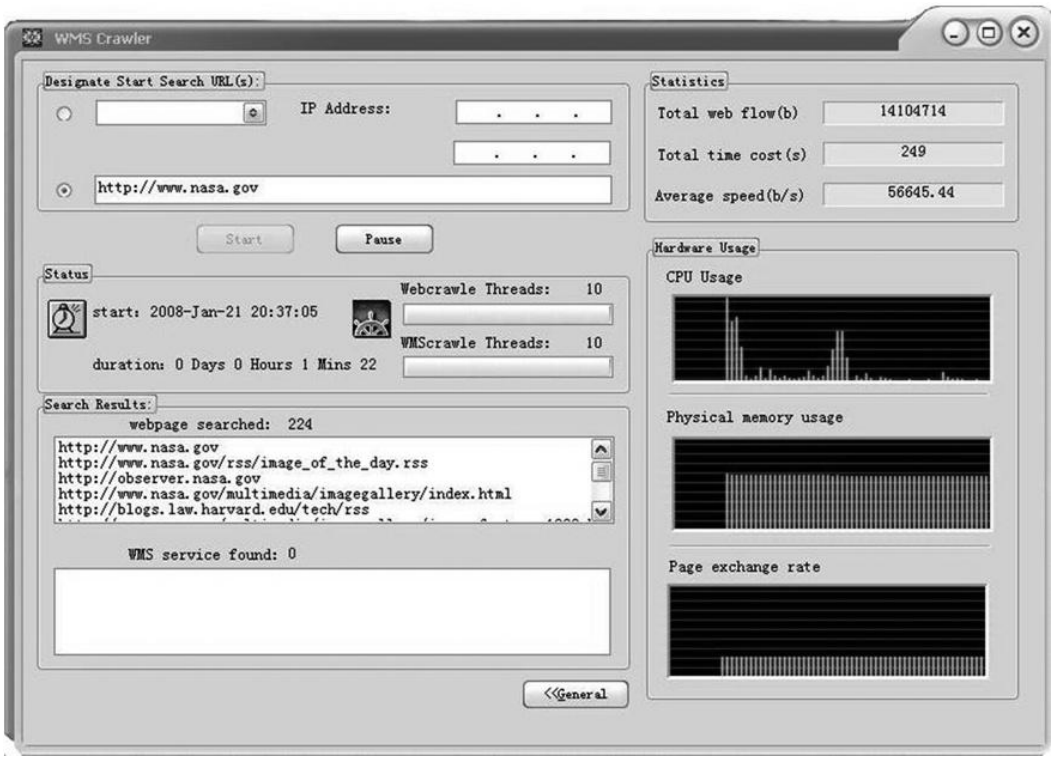


Figure 2.8. GUI of the Crawler.

The sequence diagram of the crawler can be illustrated in figure 2.9.

(1) Designate the start URL(s) and send requests to the crawler;

(2) Crawler sends HTTP GET Request to remote Web server;

(3) Return source code of the webpage to the crawler;

(4) Crawler resolves the source code, extracts all the URL addresses that exist on the webpage as hyperlinks or common text, then puts them into priority queue according to their priority;

(5) Crawler returns the accessed URL address to the client side for display;

(6) If crawler finds a similar-WMS service, sends GetCapabilities request;

(7) Capability file returns to crawler;

(8) Determines whether the URL is linked to a WMS server;

(9) Return the WMS URLs to client for displaying;

(10) Client and crawler communicate to get the status information while crawling.

For these functions, (2)-(5) are accomplished by thread group1 in modules *Sourcepage Analyzer* and *Filter*. (6)-(9) are accomplished by thread group2 in modules *R/R Handler* and *XML Resolver*.

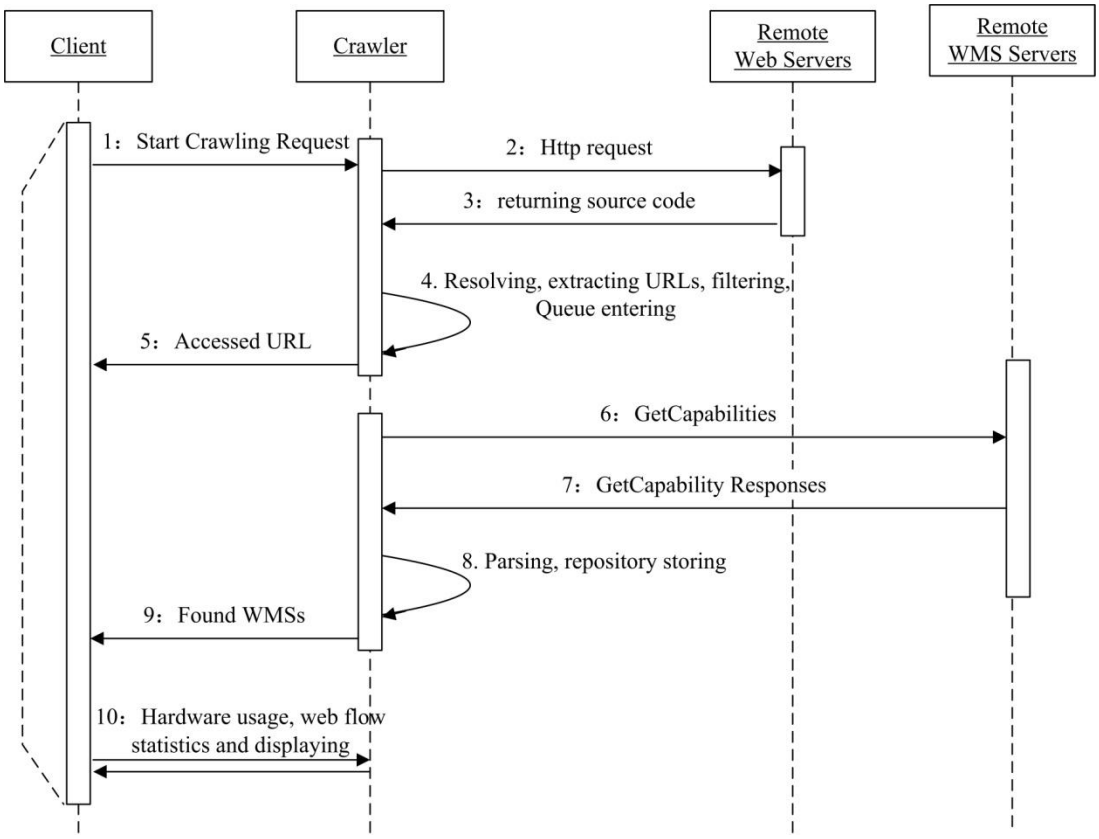


Figure 2.9. User Sequence Diagram.

2.5. Crawler Evaluation.

This Section compares the performance of the proposed crawler to that of other popular crawlers, and evaluates how well the proposed crawler performs with and without certain optional algorithms. The tests were conducted on a computer with a Pentium(R) 4 3.4GHz CPU, 4 GB of RAM and a 100Mbps Internet connection. The test includes three parts: a) efficiency b) coverage and timeliness, and 3) precision.

2.5.1 Efficiency Improvement by Concurrent Threads

Since the operating system (OS) only allocates limited CPU cycles and memory to execute a thread, it is very hard to speed up a single thread within a single program. Multithreading, which increases the utilization of a single CPU core, leverages thread-level parallelism. Especially when a thread gets a lot of cache misses, the other threads can take advantage of unused computing resources to continue the crawling task. This leads to faster overall execution and higher system throughput. However, multiple threads also interfere with each other due to the sharing of hardware resources and thus generate critical inter-thread communication overhead. The optimal number of threads is a balance between the tasks and the computing resources including CPU, memory, hard disk, and network bandwidth. The threading algorithm introduced in Section 2.3.5 generates one group of threads performing crawling tasks, and another group of threads performing determination tasks.

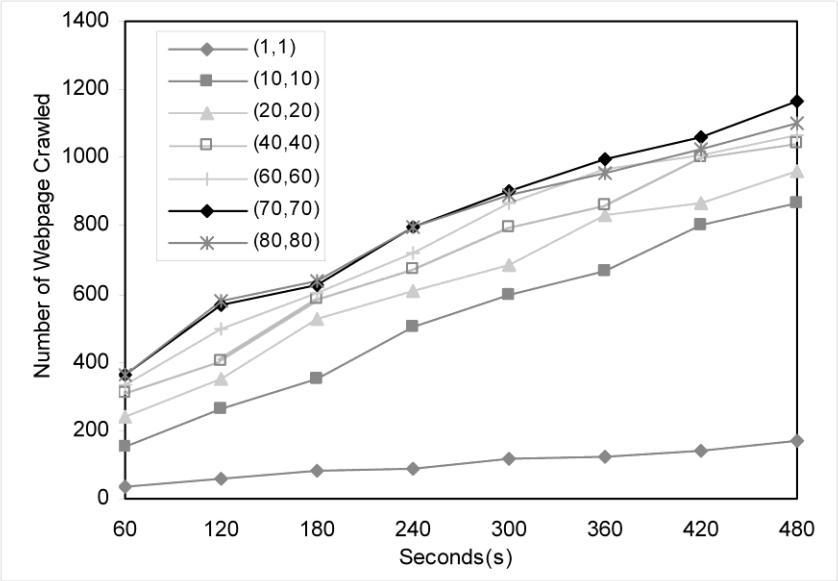


Figure 2.10 Crawling Speed with Different Numbers of Concurrent Threads.

Figure 2.10 demonstrates that: 1) with only one crawling thread and one determination thread, the crawler's throughput is very low; 2) when the number of crawling and determination threads increases up to 10 each, the crawler's throughput increases dramatically. Through successive tests, we determined that having 72 crawling threads and 72 determination threads optimized the throughput (of course, this result only applies to the tested computing environment). Equal numbers of threads were chosen for crawling and determination because the initial experiments showed that this approach optimized throughput.

It is also important that a crawler behaves politely to remote servers while maintaining a high crawling speed. If a crawler is performing many requests per second, the server performance could be negatively impacted. When calculating the crawling speed from the optimized thread groups (72,72) obtained from the above experiment, the

proposed crawler averages 0.5 seconds per page, which is consistent with the polite access interval (1-second-per-page result) proposed by Dill [79]). In addition, the priority determination mechanism introduced in the crawling algorithm helps the crawler to gather high priority pages first. Unlike general search engines that tend to crawl a webpage hosted by a server regardless of its format and content, the proposed crawler restricts its access to WMS-relevant webpages by ATF matching. Both the crawling policy and observation from real-world experiments verifies that the proposed crawler acts politely to remote web servers.

2.5.2 Coverage and Timeliness Compared to Other WMS Crawlers

Coverage and timeliness of the proposed crawler are compared to those of RR, Skylab, and GIDB (http://columbo.nrlssc.navy.mil/ogcwms/servlet/WMSServlet?). The other crawlers' results (a list of WMSs) were obtained from their official site, the duplicates and dead links were removed and non-WMS results (such as WFS, WCS or WPS) were filtered out because we are only interested in comparing WMSs. The "liveliness" of services and layers were determined by downloading and parsing capabilities of WMSs. (Term "liveliness" denotes whether the WMS referenced by a URL is actually alive and accessible.)

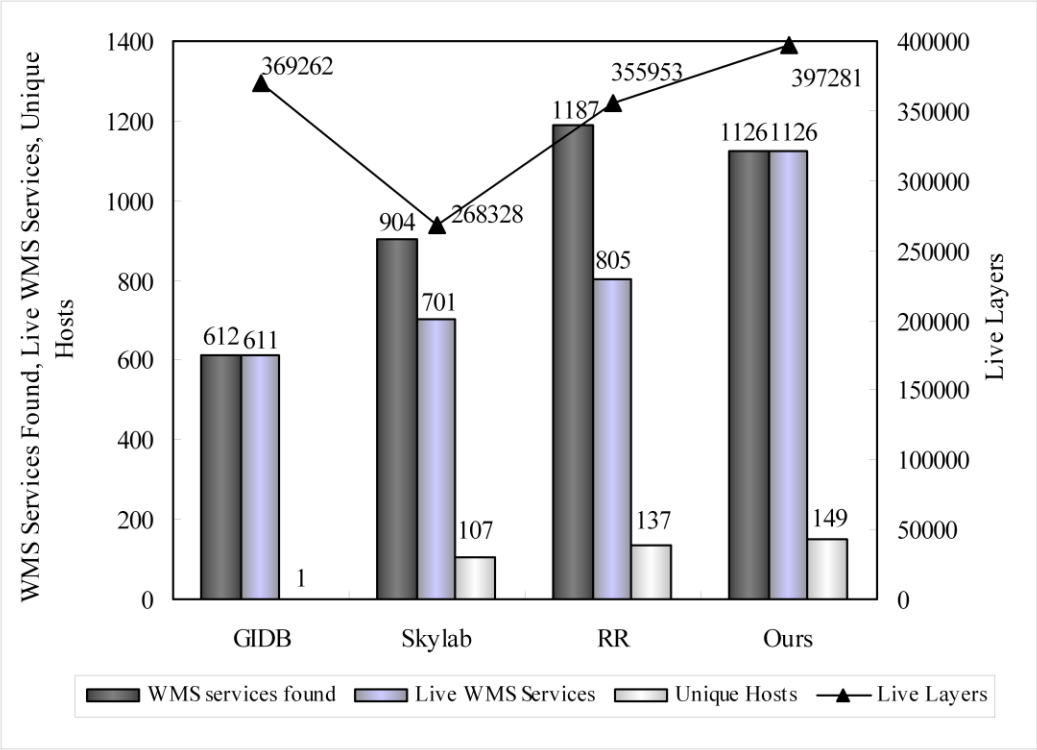


Figure 2.11 Comparison of Crawlers (GIDB, Skylab, RR and the proposed crawler) Coverage of WMSs.

Figure 2.11 shows, for each crawler, the claimed number of WMSs, the actual number of live WMSs found, the number of unique WMS hosts, and the total number of live layers. As of December 2008, the proposed crawler had the best performance, with 1126 WMSs found. In decreasing order of performance, RR found 805 WMSs, Skylab found 701 WMSs and GIDB found 611 WMSs. As RR, Skylab and GIDB rely heavily on Google's API and only match the specific format of "request=GetCapabilities" without considering other characteristic of a WMS, these crawlers fail to recognize many valid results even though they were found by Google. The proposed crawler is different in that

it takes into account the WMS web locations and the features indicated by their URLs in order to create a conditional probability based model to locate WMS quickly and accurately.

In terms of timeliness, all of the other crawlers that were tested had a significant number of dead links in their results. When employing these results into practical applications, the system performance would drop dramatically due to dead WMS reports, which is a great drawback of existing crawlers. In contrast, the liveliness of WMSs found by the proposed crawler is 100%. The proposed crawler's automatic updating algorithm, described in Section 2.3.4, enables the high liveliness rate.

Interestingly, although the proposed crawler discovered more live WMSs than the RR crawler, the number of unique hosts where the services were located was almost the same for both crawlers. An explanation is that Google (RR relies on Google API) has the advantage of a very large-scale cache that can be easily searched, even though it is not designed for domain-specific crawling. However, Google limits its crawl depth by its PR value [80], so some webpages that have higher WP or UP but lower PR will not be crawled. In contrast, the proposed crawler is better at analyzing and fully extracting WMSs from webpages in a single host. The reason for this improvement is that the proposed algorithm considers the characteristics of WMSs and employs a text analyzing technique combined with a pure hyperlink analysis.

2.5.3 Quickness in Locating WMSs and Findings Regarding WMS Distribution

The priority queue based on a conditional probability model is adopted to make the proposed crawler find WMSs more quickly. This experiment evaluates the proposed algorithm in comparison to a pure FIFO algorithm. The parameter "quickness" (3) is used to measure how fast a crawler is able to identify a WMS.

$$Quickness = \frac{Number\ of\ \ WMSs\ Found}{Total\ \ Webpage\ crawled} \tag{3}$$

Notice that the quality of seed URLs will greatly affect the experimental results. Bad seeds may lead to a very time-consuming crawling without any WMSs found. To conduct a meaningful crawl, the crawler could start from a popular geospatial website (retrieved from general search engines by the keywords provided in Section 2.3.1), or it could start from a web server that hosts a WMS or that has hosted WMSs before (as given by results of existing crawlers). Choosing one of these servers as a starting point is useful because it is more likely for such servers to host new WMS or link to other websites that host a WMS. In this experiment, the seeds that were chosen are: 1) National Aeronautics and Space Administration (NASA): http://www.nasa.gov, 2) National Oceanic and Atmospheric Administration (NOAA): http://www.noaa.gov, and 3) OpenGIS consortium (OGC): http://www.opengeospatial.org. Among these seeds, both the NASA and NOAA seed URLs are located on the crawling path of the OGC seed, which means that all the WMSs that are found by starting from NASA and NOAA's webpage can be retrieved by starting from the OGC website. Here extracting them out as independent

seeds makes the discovery process more effective by avoiding unnecessary crawling jobs. But the OGC website is still an important seed because WMSs that are not located in the NASA and NOAA's reachable network may connect from other links in the OGC website.

Figure 2.12 was generated from the experimental results of crawling the first 40,000 webpages starting at NASA, NOAA, and OGC. In those results, 23 WMSs were found. The figure shows that both algorithms have similar trends because of the FIFO algorithm utilized. However, the proposed algorithm, which applies an ATF-based conditional probability model to FIFO, is obviously faster than the pure FIFO algorithm. For example, when crawling 50% of the total pages, the proposed algorithm had found nearly 90% of the WMSs, while pure FIFO had only found 50% of those. After crawling 90% of the total pages, the proposed algorithm had found all of the WMSs, but the pure FIFO algorithm only found 90% of the WMSs. This difference is because the ATF-based conditional probability model is able to learn from past results and rank the frequencies of terms in the WMS vocabulary progressively. This way, irrelevant portions of the Web are filtered out and the crawling scope is narrowed down so that URLs that are more likely to be a WMS can be crawled earlier.

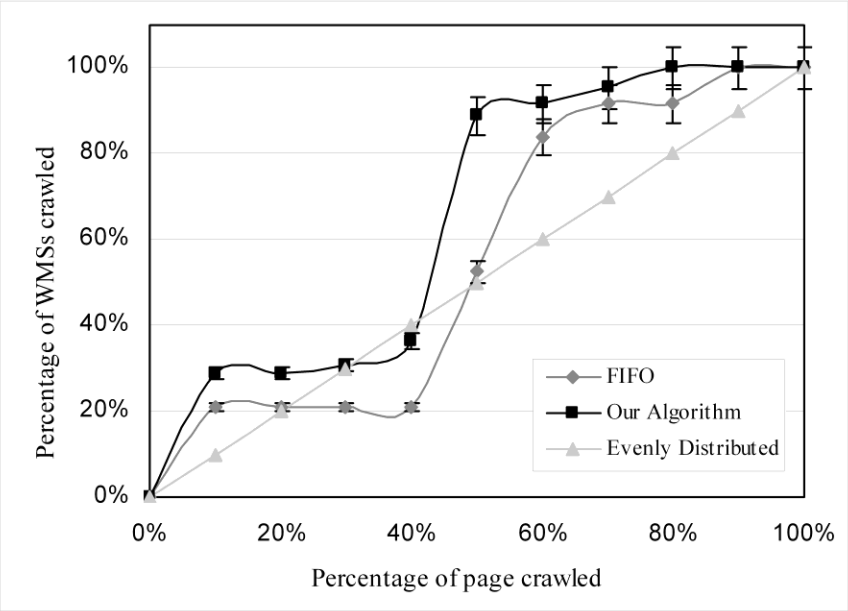


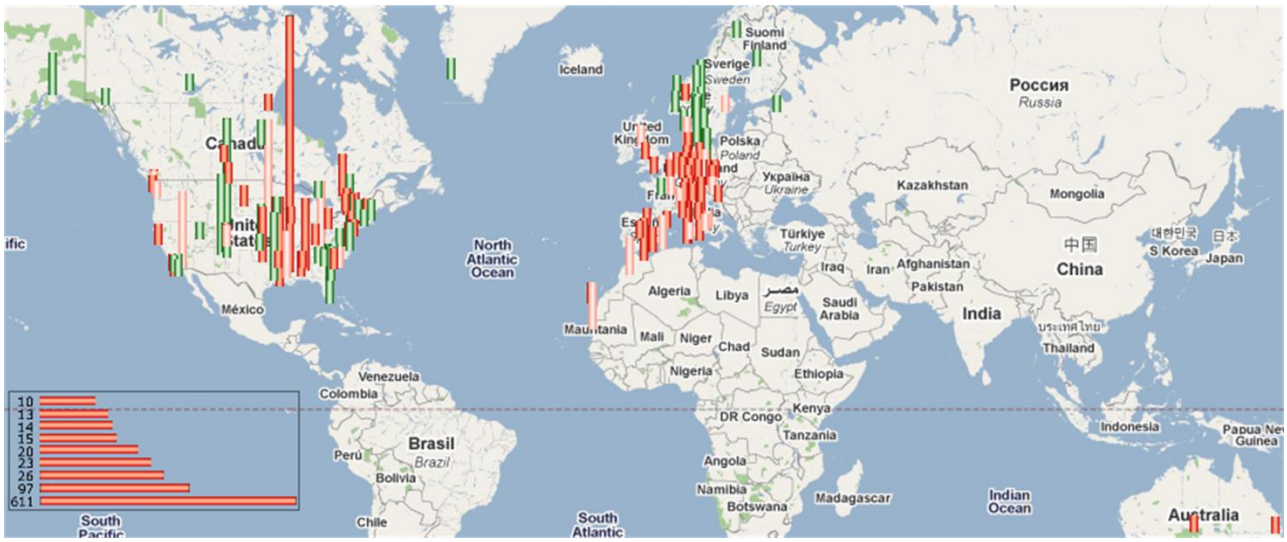
Figure 2.12 Quickness in Finding WMS by FIFO with and without ATF based Conditional Probability Model applied.

Figure 2.12 demonstrates an interesting finding in the distribution pattern of WMSs. The curves (for our algorithm and FIFO) go up quickly from 40% to 50% of crawled webpages (on the x axis), which means that the crawler quickly found a lot of WMSs (nearly 40% of all, as shown on the y axis), but from 60% to 100% of crawled webpages (on the x axis), the curve is steady and only 10% of total WMSs were found. This experiment was also conducted by crawling from other candidate seeds, such as (1) geoportal (http://geoportal.gc.ca/index_en.html), (2) the atlas of Canada (http:// atlas.gc.ca), (3) German Aerospace Center (http://www.dlr.de/en/desktopdefault.aspx), and (4) British Geological Survey (http://www.bgs.ac.uk/). All the experiments showed similar trends of WMS distribution – they are *dispersed globally and clustered locally*. It

is not hard to see that the WMSs are located in different countries, which are dispersed widely on the Web. While in a virtual place of the Web, the publishers/researchers for WMSs are usually limited to the geospatial domain and they usually link to other WMS publishers with similar topics. Therefore, a small-scale linkage graph for WMSs is formed. However if WMSs were distributed evenly, the precision curve would be a straight line as depicted in Figure 2.12.

To furthermore validate this finding, we compared both the geographic and the web distributions of WMSs retrieved by the proposed crawler and by the RR crawler. Results showed that the total 1126 (611 from GIDB and 515 from others) WMSs the proposed crawler found are scattered in eighteen countries (shown in Figure 2.13.b light red and red bars). The United States hosts the most WMSs (852), followed by Germany (145), Canada (42), Spain (38), Netherlands (12), Switzerland (6), Czech Republic (6), Australia (5), United Kingdom (4), and others (17). The WMSs found by RR (figure 2.13. a) are located in twenty countries, mainly in United States, Germany, Canada, Spain, Czech Republic, Australia, Russian Federation, Netherlands, etc. Results from both crawlers illustrated the clustering of WMSs as well. On average, the proposed crawler found eight services per server, while the RR crawler found six services per server.

a) Global Distribution of WMSs found by RR (by December 2008)



b) Global Distribution of WMS found by the Proposed Crawler (by December 2008 and June 2009)

Figure 2.13. Comparison of global distribution of WMSs found by RR and. Red bars (including light red) are the WMS distribution identified by both crawlers in December, 2008. Green bars in b) demonstrate the increased WMSs from December 2008 to June 2009. Light red bars in b) show the WMSs that were available in December, 2008 but were unavailable in June 2009. All light red, red and green bars use the same scale, shown in red in legends in a) and b). The legend in b) only shows numbers that don't exist in the legend of a). (To illustrate the results on the map clearly, the lengths of some of the bars are not to scale.)

The web diffusion of WMS servers was also observed over time (using a 6 month frame) to illustrate distribution variation. By June 2009, the proposed crawler had identified 228 more live WMSs clustered on 68 Web servers (green bars in figure 2.13 b) than found by December 2008 (Figure 2.11). The US had the biggest increase in absolute number (146). Services in Europe had increased by 78, which were distributed among Norway (60), Sweden (11), Denmark (5), France (1) and Finland (1). The crawler had also found 4 more WMSs in Canada. However, 168 WMSs (light red bars in figure 2.13 b) identified by December 2008 became dead by June 2009. This phenomenon reflects the fact that the stability cycle of a WMS is short. Although more WMSs are becoming available to provide online geospatial data services, they still lack effective maintenance methods. Meanwhile, the frequent changes of WMS hosting makes it hard for current catalogs to adapt. Therefore, these proposed crawling techniques can automatically detect the changes, the instability of WMSs creates a very challenging issue within the geospatial community.

In total, the unique web servers hosting these WMSs increased to 186 and on average there were 6.38 services per web server. These results still show that WMSs are highly clustered. Note that although the crawling task has been conducted for a long time and a significant number of WMSs have been identified, a big vacuum remains in Asian and African countries. In contrast, countries like the United States and Canada in North America and European countries like Norway, Sweden and Denmark all have significant contributions to the total WMSs. This may reflect the fact that the research of GWSs and interoperability in some Asian and African countries need great advancement.

Overall, the trends shown in Figure 2.13 demonstrate the clumped distribution of WMSs. This distribution pattern indicates that WMSs can be more effectively discovered by designing a focused and small-scale crawler than by relying on general search engines like Google. "Small-scale" is a relative measurement compared to the total size of the Web, estimated to be more than 17.27 billion (indexable) pages [81]. In total, the proposed crawler crawled around 1.3 million webpages. In comparison to Google, which uses more than 100,000 servers [82] to conduct crawling tasks, the design of the proposed crawler leads to large savings in computing resources, network bandwidth and crawling costs.

2.6. Summary

This chapter proposed a conditional probabilistic algorithm in improving crawling for quickly discovering GWSs. Multiple techniques, such as a priority queue and multithreading, are adopted to improve the crawling process in the development of the proposed crawler. The methodology, algorithm, and system architecture are described in detail. Performance evaluations are conducted from two points of view: 1) with and without improvement mechanisms and 2) comparing the proposed crawler to the three other popular GWS crawlers. Results demonstrate that the proposed crawler performs better than the others, in terms of efficiency and effectiveness, with the techniques adopted. Furthermore, the clumped distribution of WMSs was discovered: they are globally dispersed and locally clustered. Following this pattern, the Open GIS community

would be able to build a larger and more open environment for geospatial information sharing and interoperating.

Chapter 3 Semantic Reasoning: Logic Interpretation from Domain Knowledge Base

3.1 Introduction

Reasoning is the cognitive process of looking for reasons for beliefs, conclusions, actions, or feelings [83]. Based on the prior knowledge and the reasoning rules, new knowledge can be inferred [84]. The study of reasoning stems from philosophy, which studies how reasoning enables conclusions to be drawn and why some reasoning approaches are more efficient or appropriate than others. In psychology, researchers tend to study how people recognize, learn, and perform reasoning. After the invention of the digital computer in mid 20th century, "reasoning like a human" has become a long-term dream and goal for computer science and AI research. Based on insights from both philosophy and psychology, AI researchers can generate abstractions and build a computational model to simulate the human learning process on a computer and imitate human intelligence. One of the most dramatic accomplishments was the performance of the famous chess computer Deep Blue, which beat the world chess champion Garry Kasparov in 1997. After two convincing victories in 2005 and 2006, it appears that AI chess programs can now defeat even the strongest chess players [85].

Knowledge based reasoning consists of two main research branches: numerical reasoning and logical reasoning. Logical reasoning conducts inductive or deductive reasoning in explicitly defined logics using a set of inference rules. Numerical reasoning relies on statistical analysis and neurological and evolutionary theory to simulate how the human brain reasons based on existing knowledge.

3.1.1 Logical Reasoning

Logical reasoning has three sub-branches: deductive reasoning, inductive reasoning, and abductive reasoning. Deductive reasoning determines conclusions based on inference rules and preconditions. Inductive reasoning determines the rule from the KB itself, based on given preconditions and conclusions. Finally, abductive reasoning determines the preconditions that support the conclusion based on given logic rules. Logical reasoning has been extensively studied [86]-[90], but has received renewed interest as several reasoning tools have been developed to work directly on OWL files, including Racer [91], Jena [92], Fact++ [93], and Pellet [94]. These reasoning tools rely on forward chaining or backward chaining strategies to realize deduction, induction or abduction.

Forward chaining is a form of logical reasoning that progresses from what is known towards a desired solution. Using available data, a forward chaining system infers new knowledge by applying the inference rules until a goal is reached. The execution cycle is to: (1) select a rule, the preconditions of which matches the current state of the system; (2) change the system state to match the conclusions of the selected rule by executing the

rule; and (3) repeat the above steps until there is no rule to apply. Forward chaining can be categorized as a data-driven approach and the algorithm can be summarized as:

Forward Chaining Algorithm
1. Given a Knowledge Base(KB) and a goal statement (α)
2. for each sentense *s* in KB do
3. $(p_1 \wedge p_2 \ldots \wedge p_n \Rightarrow q) \leftarrow$ Standardize-Apart (*s*)
4. for each substitution θ such that $(p_1 \wedge p_2 \ldots \wedge p_n)\theta = (p_1' \wedge p_2' \ldots \wedge p_n')\theta$,
   $q' \leftarrow Subst(\theta, q)$
5. if q' is new information to the KB, add q' to New;
6. if $\varphi \leftarrow$ Unify(q', α) is not fail, then a solution is found/validated, return φ
7. Merge KB $\leftarrow$ KB $\cup$ New
8. Loop steps 2-7 if the set of new inferred knowledge(New) is not empty, else return false

As an example, suppose that the goal is to ascertain the color of a pine tree. Given that pine is a woody plant with branches, assume that the rule space contains the following rules:

Rule 1: $Plant(X) \wedge Woody(X) \wedge Has(X, branches) \Rightarrow Tree(X)$
Rule 2: $Plant(X) \wedge$ Flowering(X) $\Rightarrow Flower(X)$
Rule 3: $Tree(X) \Rightarrow Green(X)$

Rule 1 would be searched because we can find satisfactory substitutions (X=pine) for the conditions based on the given data. Then new information $Tree(pine)$ is added to the KB. Next, Rule 3 is selected because its antecedent $Tree(X)$ matches the knowledge we inferred. Now a new consequent $Green(pine)$ is added. No more knowledge can be inferred from this information, but through unification, we can accomplish our goal of determining the color of pine, which is green.

In contrast with forward chaining, backward chaining starts with the goal to be achieved and repeatedly breaks it into sub-goals that are easier to solve with the available data and the inference rule space. An inference engine with a backward chaining algorithm continues to search the inference rules to find one with a consequence matching the current goal. If the precondition of that rule cannot be confirmed to be true using the existing KB, the precondition must be added to a list of unsolved goals (sub-goals), which should be validated using other rules and data. The backward chaining algorithm is:

Backward Chaining Algorithm
1. $BC(KB,$ goals, $\theta)$
2. *if* empty(goals), then return $\{\theta\}$
3. $q' \leftarrow Subst(\theta, fisrt(goals))$
4. for each sentense s in KB, where
   Standardize-Apart $(s)=(p_1 \wedge p_2 ... \wedge p_n \Rightarrow q)$ and
   $\theta' \leftarrow$ Unify(q, q') succeeds
5. ans $\leftarrow BC(KB,$ rest(goals), $\theta \bigcup \theta')$
6. return *ans*

Backward chaining is an iterative process. In the above algorithm, *ans* is a set of satisfied substitutions from which all goals and sub-goals can be achieved. For the same pine color example, the goal in backward chaining is to find evidence (knowledge) to support $Green(pine)$. First, Rule 3 is searched and selected from the rule space because its conclusion matches the goal to determine the pine's color. As the condition ($Tree(pine)$) of the goal $Green(pine)$ is not supported directly by the available data, it is inserted in the goal list as a sub-goal. The rule space is traversed again and then Rule 1 is

selected. The preconditions of Rule 1 match the available data $Plant(pine), Woody(pine)$ and $has(Pine, Branches)$; therefore, a chain of reasoning demonstrating that the color of pine is green has been established using Rules 1 and 3 and the given knowledge.

3.1.2 Cognitive Reasoning

In addition to logic learning, cognitive learning, which employs theories from AI, serves as an effective tool to learn from data and to explore implicit knowledge and regularities. Typical cognitive reasoning paradigms include genetic algorithm, reinforcement learning, etc.

The genetic algorithm (GA) [95]-[96] is an important branch of numerical reasoning, which uses models from biological evolution to guide computer simulations. It is an automatic reasoning process used to find exact or approximate solutions to global optimization problems. A typical GA algorithm works as follows:

Generaric Algorithm
1. Generate a random initial population
2. Evaluate the fitness of individuals in the population,
   if the ideal individual is found, finish and exit.
3. Initial an empty population *P*,
4. SELECT individuals *i* and *j* from old population
5. CROSSOVER between *i* and *j*
6. MUTATE *i* and *j* seperately and add new *i* and *j* to *P*
7. if P is not full, go loop 4-6, else replace old generation with *P*.
8. go to 2

Initially, a KB is represented by the population and each fact in the KB is an individual. A fitness function is used to measure how well the individuals achieve the

goal. In other words, has the learning process entailed the exact knowledge we want? This learning process is carried out by iterative selection, crossover, and mutation, in analogy to biological adaptation and evolution. Among these operators, selection is used to determine which individuals will be chosen for later breeding. Available selection algorithms include fitness proportionate selection (FPS) and tournament selection (TS).

FPS selects an individual based on the distribution of possibility, which equals the normalized fitness value of individuals. Thus, the higher the fitness of an individual, the greater the chance it will be selected. In contrast, the fitness value in TS does not dictate the selection mechanism. In TS, N individuals are picked randomly and the most fit individual is selected. A value of N that is too small will make the system wander aimlessly and find a solution very slowly. A value of N too large will reach a solution quickly, but it might be suboptimal. In practice, N=7 is a proper size to balance out the above effects.

Another process for genetic-based learning is crossover, which allows sexual reproduction (gene modification among more than one individual). Typical algorithms for crossover include one-point crossover, two-point crossover and "cut and splice." In a one-point crossover, two individuals are broken at the crossover point and the pieces are swapped to generate new individuals. In two-point crossover, the individuals are cut into three pieces, and the middle piece is swapped. In a third crossover method "cut and splice," two individuals have different crossover points. Compared to the previous two crossover algorithms, cut and splice will lead to different lengths in the newly generated

individuals. Another method of reproduction is mutation, which changes an arbitrary bit in a gene sequence (individual) to enhance the diversity and reduce the similarity in individuals. Without mutations, the system tends to become fixed at a local optima and does not continue learning.

Reinforcement learning is a type of numerical reasoning that learns new strategies to guide how a computer agent takes actions in an environment. The environment is essentially a Markov model, containing a set of states (S) connected by transitions or actions (A), and a set of scalar "rewards" (R). Each time an agent performs an action, a state transition occurs and a reward (reinforcement) is received (which might depend only on the current or previous actions). The learning process fills up a reward table of a Markov model in which the number of rows equals the number of total states and the number of columns equals the number of actions. Each cell of the table stores the reward value for each possible action in that state. If the agent takes an action i in state S_j, it will get a reward R_{ij} by taking the chosen action. The reinforcement learning process can be described mathematically as: at each state s, the intelligent agent will choose the best action, represented by the highest reward (based upon the current move plus any expected future rewards). The detailed algorithm is as follows:

Reinforcement Learning Algorithm

1. Intialize the reward table ($n \times m$) which includes every action (m) at each state (n).
2. Loop for each move of the intelligent agent:
3. S $\leftarrow$ initial state of the agent
4. For each time the agent needs to make a decision on where to move
   always try to best action
   $Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha(r + \gamma \max_{a'}(s', a'))$
   $s' \leftarrow s$
5. Return reward table

The above learning process uses an active learner that allows modification of the policy on the fly as the intelligent agent learns. The computer agent can become more intelligent by gaining new knowledge through continuous learning.

3.1.3 Comparison of Logic Reasoning and Cognitive Reasoning

Table 3.1 demonstrates the advantages and disadvantages of using logical reasoning and cognitive reasoning, where logic reasoning can be categorized as declarative methods (requires extensive reasoning at run time) and cognitive reasoning can be categorized as direct methods (requires no reasoning at run time). In cognitive reasoning, all the assumptions, criteria and goals are determined before "running time," whereas, in logical reasoning, these factors can be changed dynamically during the run-time inference. For example, the learned knowledge can be factored into separate groups, used for reasoning in different branches and then recombined when necessary. Besides, cognitive reasoning is a statistical process that learns knowledge from only a sample set of the full knowledge space, therefore, the error is inevitable. Logical reasoning also works from a limited size of knowledge. However, by defining rules and regularity, the deducted knowledge tends

to be more accurate. Plus, the schemes of logical reasoning have been much researched and a set of formal languages, ontological editing and reasoning tools have been developed to support fast prototyping; therefore, it is chosen as the major methodology to support the knowledge discovery in this dissertation.

Table 3.1 Comparison of Logical Reasoning and Cognitive Reasoning.

Strategy	Advantages	Limitations
Logic Reasoning	1. A large pool of formal languages to choose for representing the logic. 2. Knowledge structures are easier for people to understand and validate. 3. Modern ontology languages offer well-proven, standardized representation and reasoning mechanisms 4. Mature ontological editorial and reasoning tools to support fast prototyping for various applications.	1. Hard to handle uncertainty 2. Needs to sacrifice expressivity to achieve high reasoning ability.
Cognitive Learning	1. Performs efficiently in the presence of uncertainty 2. Skillful at solving optimization problem.	1. Has no consistent format for knowledge organization and presentation. 2. Studies only the simplest forms of experience for learning, distant from people's daily reasoning procedures.

3.2 Logic Basis for Knowledge Representation

Knowledge representation refers to the general topic of how information can be approximately encoded and used in the computational models of cognition [97]. In practice, it is impossible to represent everything in the world. However, we are able to

represent the knowledge in a precise way that when a conclusion is drawn from the available information, it is guaranteed to be correct. Knowledge representation is a substitute to the real world knowledge and the way it models the world could be from any perspective. Therefore, selecting a representation in terms of the concepts, properties and interrelations means making a decision on how and what to see in the world [98].

A representation language defines both the syntax and semantics to validate a sentence. Syntax refers to the grammatical structure, which specifies that a sentence is well formed. Semantics, on the other hand, defines the truth of each sentence in a domain. A sentence that is syntactically correct does not necessarily mean it is also correct semantically. For example, the combination of <subject predicate object> in this exact order declares the syntax. Therefore, the sentence <apples flow fast> is a well-formed sentence. However, this sentence does not make sense because it is semantically invalid. In texts, there are many paradigms for formalizing knowledge representation, such as PL, FOL, and Description Logic (DL). In the next section, the fundamental basis for each logical language is introduced.

Logic enables a precisely formulated subset of language to be expressed in a computable form [99]. Logic forms can be decorated with word senses to disambiguate the semantics of the word [100]. Three forms of formal languages have been used in AI to model the general laws of truth: PL, FOL and DL.

Propositional logic (PL), which is also known as boolean logic, is a branch of logic that studies ways of combining or altering statements or propositions to form more

complicated statements or propositions [101]. There are two symbols with fixed meaning, "true" is an always-true statement and "false" is an always-false statement. Other non-always-true statements are combined from simpler statements using connectives. Table 3.2 illustrates the five connectives used in propositional logic. $\neg$ means negation; the negation of a true statement is false and the negation of a false statement is true. $\wedge$ is used to represent conjunction and a statement which has $\wedge$ as the main connective is also called an and-statement. An and-statement, e.g. $A \wedge B$, is true only when both A and B are true. If either A or B is false, the and-statement is false. $\vee$ is used to represent disjunction and a statement which has $\vee$ as the main connective is called an or-statement. An or-statement is true when any of the disjuncts are true and an or-statement is false when all of the disjuncts are false. $\Rightarrow$ is a symbol for implication; it represents an if-statement, meaning that when the first part of the statement is true, then the second part can be implied as true as well. $\Leftrightarrow$ connects an if-and-only-if statement, meaning either part of the statement cannot be true without the other. Through the above five connectives, complex sentences can be constructed and some basic reasoning tasks can be conducted by enumerating the truth values for all the atomic sentences as parts of a complex sentence.

Table 3.2 Connectives in Propositional Logic.

Connectives	Meaning
$\neg$	Negation
$\wedge$	Conjunction
$\vee$	Disjunction
$\Rightarrow$	Implication
$\Leftrightarrow$	Bicondition

Propositional logic has sufficient power to express partial information, enabled by disjunction and negation. However, it lacks the expressive power to describe a situation that includes many objects concisely. For example, it is quite easy to describe "people $\{P_1, P_2 \cdots P_n\}$ with grey hair $\{H_1, H_2 \cdots H_n\}$ are seniors $\{S_1, S_2 \cdots S_n\}$" in natural language. P_i is the person i, H_i refers to the statement of "the hair of P_i is grey" and S_i refers to the statement that P_i is a senior person. The statement is very hard to express using PL. One has to write a separate rule for this statement, such as

$$(P_1 \wedge X_1) \Rightarrow S_1$$
$$(P_2 \wedge X_2) \Rightarrow S_2$$
$$\cdots$$
$$(P_n \wedge X_n) \Rightarrow S_n$$

Another form of formal language, FOL, which has been studied for many decades, has sufficient expressive power to represent commonsense knowledge. FOL assumes that the world not only contains facts, but also objects (people, animals, buildings, etc.), relations (has color, bigger than, etc), and functions (is the teacher of, the end of, etc.). It actually decomposes the statement into smaller granularity to encompass higher expressiveness. Therefore, FOL contains more elements than PL, as shown in Table 3.3:

Table 3.3 Basic Elements of FOL

Elements	Example
Constant	David, red, Mississippi River
Predicate	More than, deep than
Function	Sum, is mother of
Variable	a,b,c,x,y,z
Connective	Same as those listed in Table 3.2
Quantifier	$\forall, \exists$

For predicate symbols, statements can be written as Morethan(2,1). This is an atomic sentence with true or false as its value. For function symbols, the objects they refer to can be used without their names but with any variables. For example, in $sum(x, y) \Leftrightarrow x + y$, it is not necessary to give a value for x and y at the time to define the function "sum." In addition to predicates and functions, there are two quantifiers defined in the first-order language: the universal quantifier $\forall$ and the existence quantifier $\exists$. The usage of universal quantifier is $\forall$<Variable><Sentence> .For example, $\forall x\ River(x) \Rightarrow flowsInto(x,\ Sea \vee Ocean)$ means that for all the objects that are rivers, they all flow into the sea or ocean. This expression can be converted to the conjunction of instantiations of valid variables,

$$\begin{aligned} & River(Mississippi) \Rightarrow flowsInto(Mississippi,\ Sea \vee Ocean) \\ \wedge\ & River(Yellow) \Rightarrow flowsInto(Yellow,\ Sea \vee Ocean) \\ \wedge\ & \ldots \end{aligned}$$

The existential quantifier $\exists$ expresses the partial relationship. A sentence

$$\exists x\ River(x) \Rightarrow flowsInto(x,\ Ocean)$$

is equivalent to the disjunctions of instantiations of valid x:

$$\begin{aligned} & River(Yangze) \Rightarrow flowsInto(Yangze,\ Sea) \\ \vee\ & River(Yellow) \Rightarrow flowsInto(Yellow,\ Sea) \\ \vee\ & \ldots \end{aligned}$$

The above universal instantiation (UI) and existential instantiation (EI) can be used to infer implicit knowledge from existing ones. The UI can be applied to add new sentences to the KB and the new KB is logically equivalent to the old one. EI can be

applied one at a time to replace the existential sentences. The new KB is not logically equivalent to the old one but is satisfiable if the old KB is satisfiable.

The syntax and semantics defined in FOL make it easy to describe facts about objects. Although FOL is very expressive in modeling the knowledge, the computational complexity exceeds high polynomial to exponentially time. FOL is also bad at handling default information, which leads to inconsistency of the KB. For example, there is a FOL rule:

$$\forall x \; bird(x) \Rightarrow flies(x)$$

Known that $bird(opus)$, $bird(tweety)$ and $\neg flies(opus)$, the KB becomes unsatisfiable. To restrict the syntax of FOL and to reduce its computational complexity, a higher-order logic – DL is suggested. DL is the subset of FOL with limited expressiveness and more decidable logic. The restrictions that are added include allowing the negation of any object, not just the atomic objects. The language formed by adding the above restriction is a centrally important DL, called $\mathcal{ALC}$. $\mathcal{SHIQ}$ is another popular DL and it extends $\mathcal{ALC}$ by adding cardinality restrictions to quantify predicates, and transitive and inverse predicates. $\mathcal{SHOIN}^{(\mathcal{D})}$ is another widely used DL based on $\mathcal{ALC}$; it supports the use of datatype properties, data values and data types. $\mathcal{SHOIN}^{(\mathcal{D})}$ also supports the enumerated classes of object value restrictions, such as "OneOf" or "hasValue." $\mathcal{SHOIN}^{(\mathcal{D})}$ provides a tradeoff between the decidability of reasoning tasks and its associated computational complexity. It is, therefore, used as the supporting logical language for building the KB in the dissertation.

3.3 Build-up of a Hydrology KB

Previous efforts for knowledge representation focused mostly on the syntax level. To make the machines automatically understand and process a user request for conducting various reasoning tasks, the knowledge representation also needs to be considered at the semantic level, where data are annotated semantically and the machine possesses reasoning capability. A use case for water community is: scientists and researchers want to monitor how melting of snow cover and solid ice influence the bio-habitat in the Arctic region by a single simple query. This requires building a domain KB consisting of: (1) rich science keywords such that any associated datasets can be retrieved by semantic analysis rather than simple keyword match (to increase "recall" of an IR task); (2) rich domain knowledge such that similar spatial concepts can be distinguished (to increase "precision" of an IR task).

3.3.1 Knowledge Base

A KB, also called an ontology, encodes the explicitly defined formal specification for a shared conceptualization [102]. Use of an ontology helps to discover the implicit relations between concepts that are not usually made explicit in traditional databases [103]. Several hydrology ontologies have been developed to serve the needs of the broader hydrology community. The CUAHSI (Consortium of Universities for Advancement of Hydrologic Science) developed a taxonomy-based ontology. It focuses on classifying the key components (e.g. precipitation, radiation and WaterBody) in the water cycle and the interaction of the hydrosphere with the atmosphere and the biosphere

[104]. Another source of hydrological knowledge is derived from the NASA's Global Change Master Directory (GCMD) [105] keyword collection. The GCMD contains 1000 controlled keywords used by clearinghouses to classify Earth Science resources. An additional 20,000 uncontrolled keywords in climatology, marine, geology, etc. were extracted from the descriptions of the data and service providers. Other existing data dictionaries for environmental knowledge modeling include General Multilingual Environmental Thesaurus [106] and the INSPIRE hydrological theme initiatives [103].

The taxonomy and controlled keywords provide valuable guidance to distinguish terms; however, these efforts contain few interrelation and association definitions. To overcome this issue and make the available terminologies maximally reusable, Earth scientists developed SWEET [36] to model scientific terminologies and their interrelationships. The modularized ontology SWEET 2.0 builds upon basic math, science, and geographic concepts to include additional modules for the planetary realms, such as Hydrosphere, Cryosphere, Atmosphere, Geosphere, Biosphere, etc. This modularized design facilitates the domain specialists to build self-contained specialized ontologies that extend existing ones. It is the upper-level guidance for building up the hydrological ontology in this dissertation, as the top layer of Figure 3.1 shows.

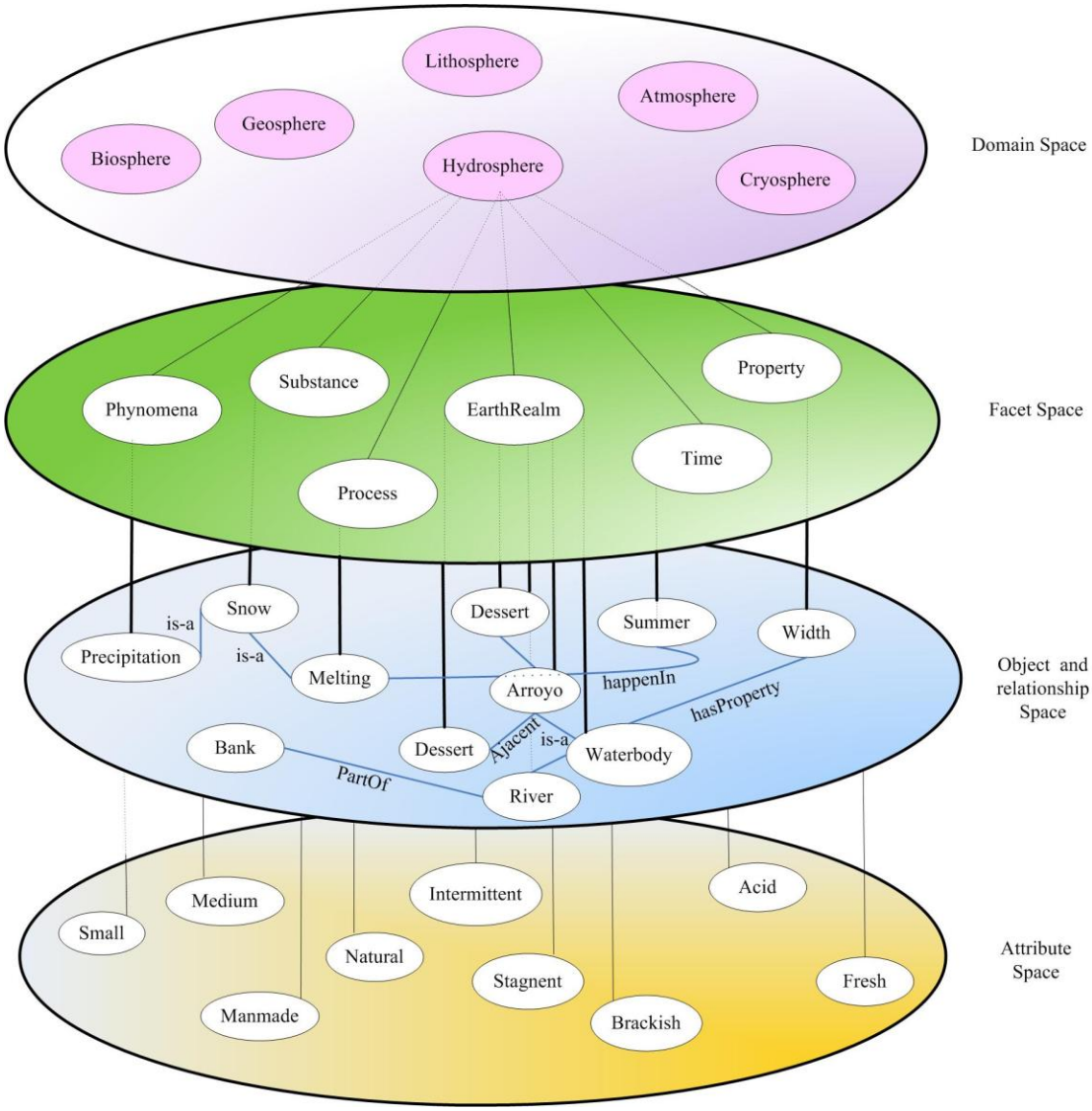


Figure 3.1 Conceptual Model of a Hydrology Ontology.

Figure 3.1 demonstrates the conceptual model in terms of various spaces. To model the domain KB, a conceptual model is needed to help to understand the problems and its constituents, and the way how reality is composed and represented [107]. The first step to build a conceptual model is to break down the domain into atomic components and then

relate them using a formal language. As Figure 3.1 shows, the Earth Science domain can be decomposed into several sub-domains, as emphasized in the SWEET 2.0 above.

Within each domain, the terminologies are mapped into facets: (1) Phenomena, encoding any observable occurrence that belongs to a domain; (2) Substance, encoding non-living building-blocks of nature; (3) Earth Realm, encoding the "sphere" or environments of the Earth; (4) Process, encoding the change or conversion that has happened; (5) Property, encoding the attributes of the terminologies associated to other facets [36]. "Phenomena," "Substance," "PlanetaryRealm," "Process," and "Property" compose the facet space, shown as the second layer of Figure 3.1. Whereas the top two layers contain the abstractions of domains and facets within a domain, and the bottom two layers contain more specific, real world terms, which provide knowledge for building the KB. For a hydrology ontology, categorizations of terminologies and the relationships among the terminologies are defined in the "object and relationship" space. For example, we can define super-sub class containment "Snow" is a type of "Precipitation" and "River" is a type of "WaterBody." We can also define relationship "partOf" or "hasAjacentLandform" to represent that "Bed" is part of a "River" or "Arroyo is always formed in desert area." The knowledge defined in this layer makes the logic reasoning possible (discussed in Section 3.4). However, when humans want to identify the similarity or differences between the objects, the knowledge about the properties of each object is required. To capture the essence of human perception, dominant attributes of objects need to be collected and modeled. This part of knowledge is defined in the attribute space (bottom layer of Figure 3.1).

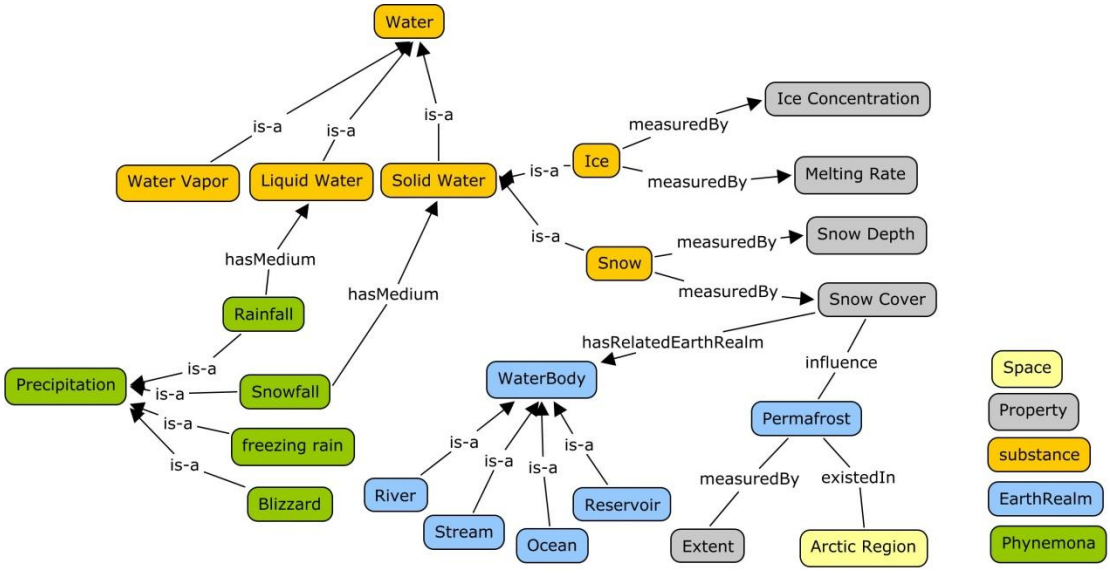


Figure 3.2 Ontology Fragment Encoding Hydrology Knowledge for Arctic Research.

To provide a more specific picture of object and relationship space, an ontology fragment is extracted from the hydrology KB to show the interrelationships between concepts that could be used for ontological reasoning, as shown in Figure 3.2. The object nodes with different colors mean the mapping from lower layer into upper facet level. All of the terminologies in each facet are encoded into one OWL file and these files are connected with each other using <owl:imports> provided in the W3C OWL standard. In total, the hydrological KB used for reasoning work in this dissertation is around 3000.

In Figure 3.3, an attribute space to describe terminology “WaterBody” defined in Figure 3.2 is illustrated. The notations on the arrows connecting the “WaterBody” node and all the green nodes are the list of attributes for describing the object “WaterBody.” Other nodes that are descendants of the green nodes are the objects that a “WaterBody” has with a certain predicate. The set of functions that a WaterBody has is {Irrigation,

PublicSupply, Recreation, PassengerExchange, EcologicalFlow, PowerPlant, Industry, Building/RepairingBoats, Wildlife, Aquaculture, FloodProtection, Mining, LiveStock, ShipShelter, HydroElectricPower, TransferingCargo, WaterQualityImprovement, ShorelineErosion, and Aesthetics}. Each member in this set can used to replace "Function" in the triple expression {"WaterBody," "hasFuntionality," Function}. For example, the triple {"WaterBody," "hasFunctionality," "PublicSupply"} means that a type of "WaterBody" can be used for supplying water to the public. Any object that is a subclass of "WaterBody" can be applied to the framework with the specific attribute value defined for that object. For a WaterBody "River," property "FlowingRate" of it is "Flowing" rather than "Stagnant"; for a WaterBody "Lake," property "FlowingRater" of it is "Stagnant." After modeling each WaterBody object, the similarity judgment or recognition can be conducted (discussed in Chapter 4).

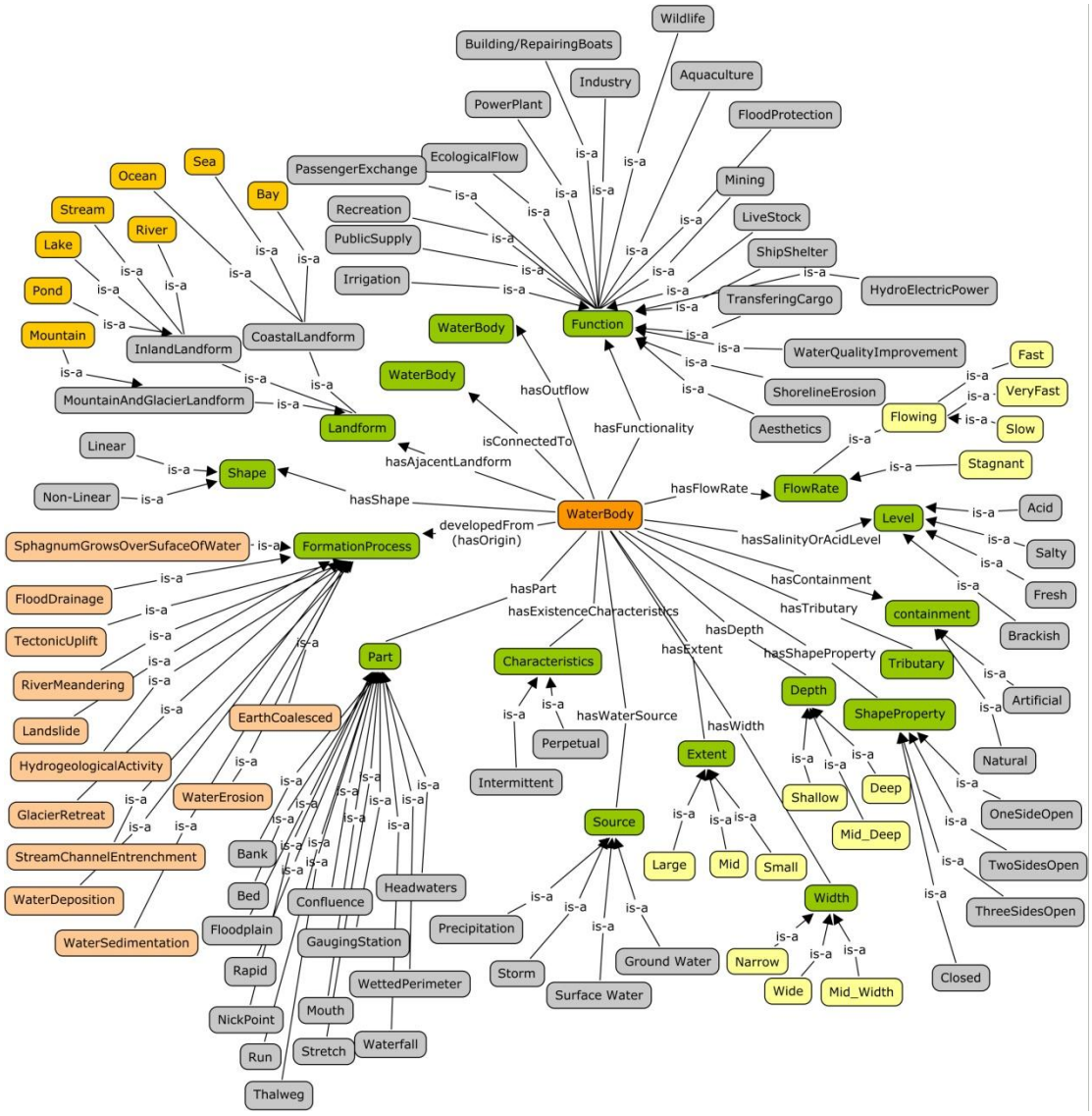


Figure 3.3 An Attribute Space for “WaterBody”.

3.3.2 Development of a Knowledge Base

Semantic registration technique is an important technology to develop a KB from the conceptual model. It works by extracting the meanings of unstructured documents, expressing them logically, and then storing them into a domain KB. The goal of semantic registration is to find patterns in the semantic-lack document and use these patterns to

support semantic reasoning and ranking. To achieve this goal, the formats of information sources need to be analyzed. There are two major types of information sources: news in an unstructured webpage and metadata in a structured XML format. The web contains a huge amount of useful information and the content of the information has real-time characteristics. People rely heavily on the Web to absorb and digest knowledge; therefore, using information from the Web and providing a tool that allows a Web user to input the knowledge is selected to be the major methods for semantic registration.

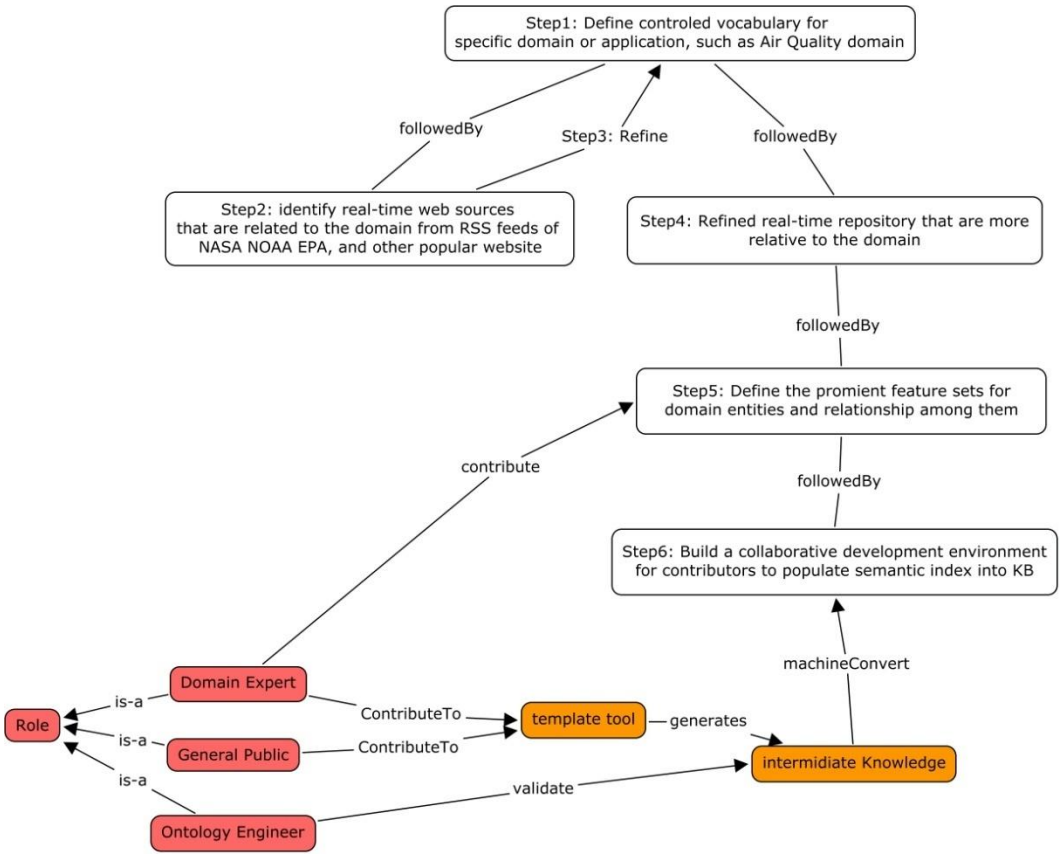


Figure 3.4 Development Workflow.

Figure 3.4 demonstrates a scenario for collaboratively populating a domain ontology. The first step is to analyze the problem and the type of available data sources, and create data-to-be controlled vocabularies for the domain. Using a controlled vocabulary, identified real-time sources can be filtered and only the closely related sources are selected manually or automatically by machines. The next step is to develop a semantic schema consisting of a set of properties that are prominent to represent objects and the logical relationships among them. The conceptual model (Figure 3.1) provides the guidance for collecting the objects, the attributes of the objects and the interrelationships of the objects within a facet and across the facets. Therefore, it can be used as the semantic schema for building the hydrology ontology.

Semantic Web has an AAA slogan, namely "Anyone can contribute Anything at Anytime to the semantic Web". Towards this end, we aim to provide a collaborative development environment to make the KB available to a broad audience in the Earth Science community. Using this tool, any researcher can create, evaluate, access, and reuse ontology easily. This idea is generated from the Google directory, where anyone can become an editor and help categorize products, information, etc. We believe that the more people who become involved, the quicker the KB can be enriched. Figure 3.5 shows the architecture of the Web-based tool for the collaborative ontology development.

For users with different backgrounds to be more actively involved in the development of KB, three types of GUIs are provided. For the general public, a template-based interface is provided. They do not need to understand the technical details

of how KB is structured and how to encapsulate knowledge using a formal language; instead, they can directly input the knowledge based on the template generated automatically according to the backbone ontology schema. An ontology engineer can operate both the semantic schema and the ontology using an online ontology editor – Web-Protégé. In contrast, an ontology expert can register the ontology fragment encoded in formal language (such as RDF or OWL) directly to the backend KB. Through this role partition and the provision of the Web-based tool, anyone who is interested in hydrology ontology development can contribute to it.

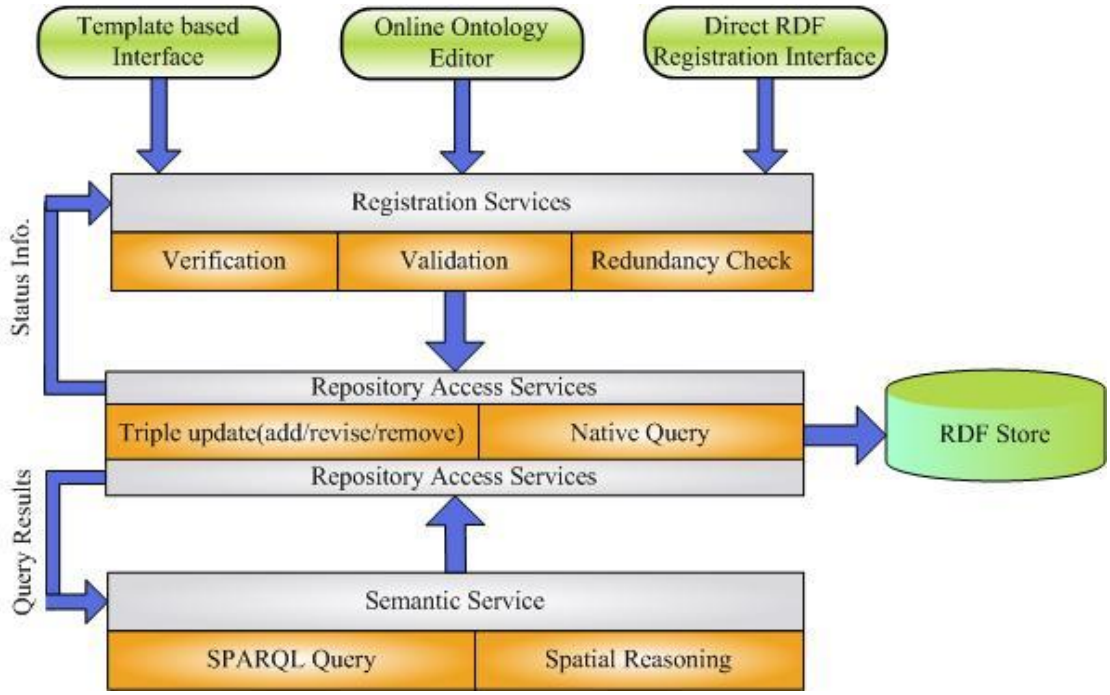


Figure 3.5 Architecture of the Collaborative Ontology Development (COD) Tool.

Figure 3.5 shows the architecture of the Web-based ontology development tool. There are three types of services: registration service, repository access service, and

semantic service. Registration service is responsible for verifying, validating, and checking the redundancy of the knowledge, which was input by remote users. The verification and validation is essentially the same as the syntax checking. To avoid redundancy of the backend repository, only the validated metadata will be registered to the repository by the mediation of the registration services. The repository access services are responsible for triple operations such as adding, removing, or updating the KB. Meanwhile, they also handle the semantic queries (instance query, sub class-super class query or any other user defined queries) sent from semantic services. By continuous contributions from the whole Earth Science community, the KB will be enriched, extended, and served as a large inventory to support semantic reasoning for various applications. The semantic services will be discussed in details in the following sections.

3.4 Knowledge-based Semantic Reasoning

Semantic reasoning is the core component of the semantic search engine [108]. Given a user query, syntax analysis, semantic analysis, and IR tasks from a heterogeneous environment are performed in sequence [109]. Syntax analysis focuses on analyzing components of a query sentence, as well as getting the ‘central word,’ i.e. the exact object that interests the user. This type of analysis would help us efficiently retrieve the ontology and proper candidates. Syntax analysis can be conducted by either providing a query template for a user to map phrases into different dimensions, e.g. “WHAT,” “HOW,” “WHEN,” and “WHERE,” provided in a GUI, or relying on a natural language

parser, such as the Stanford open-source statistical parser [110]. In this paper, a template-based approach is chosen to save client parsing time.

After syntax analysis, a user query is mapped into two levels: logic and formal query levels for semantic reasoning. When reasoning is conducted, complex queries will be decomposed into sub-queries. For example, suppose a researcher wants to study, “How does solid water melting influence stream flow in the Arctic Region over the summer?” Through syntax analysis, the “solid water” can be distinguished as an event, which is the central word of the whole sentence, “melting” as the state change process, “Arctic” as a place, and “summer” as a time. Given this information, the natural language query can be transformed to a DL-based query for machine reasoning:

Q1: $\text{SolidWater} \cap \exists \text{hasProperty.Melt} \cap \exists \text{hasObject.Stream} \cap \forall \text{takePlaceIn.Arctic} \forall \text{hasTime.Summer}$

By iteratively unfolding Q1 from the central word, “Solid Water,” information which is considered more useful can be retrieved based on the knowledge encoded in the ontology. This process is called query decomposition. Given the ontology fragment provided in Figure 3.2, Q1 could be decomposed into:

Q1a: $\textbf{SomeSWClass} \cap \exists \textit{isSubClassesOf}.\text{"Solid Water"}$

Q1b: $(\text{AProperty} \cap \exists \textit{isSubClassOf}.\text{"Property"})$
$\cap (\text{AProperty} \cap \exists \textit{is}\,\text{Pr}\textit{edicateOf}.\text{SomeSWClass})$
$\cap (\textbf{Parameter} \cap \exists \textit{isObjectOf}.\text{SomeSWClass})$

Q1c: $\textbf{SomeStreamClass} \cap \exists \textit{isSubClassesOf}.\text{"Stream"}$

Q1d: $(\text{Parameter} \cup \text{SomeStreamClass})\textbf{.hasData} \cap \forall \textit{takePlaceIn}.\text{"Arctic"} \forall \textit{hasTime}.\text{"Summer"}$

In this query, Q1a and Q1c aim to find <n1, n2…nk> of all of the subclasses and other related terminologies of the given terms. This type of inference could be considered as a query expansion process on the class level, because terminologies with similar meanings but not designated as a query term could be provided. From the ontology provided in Figure 3.2, {“Snow” and “Ice”} will be returned as the set of “SomeSWClass” for Q1a and {“River” and “Creek”} will be returned as the set of “SomeStreamClass” for Q1c. Q1b is formed by checking all of the roles that are connected with “SomeSWClass” and the connected predicate is a type of “Property.” “AProperty” is the intermediate variable and “Parameter” is the variable for the expected results.

Given the KB above, “Ice Concentration,” “Snow Cover,” and other parameters that are used to measure the variation of snow and ice are returned. Compared with Q1a, Q1b, and Q1c, which infer results within the scope of the KB, Q1d can be considered as an external search. After the desired variables in Q1a-c have been inferred, Q1d redirects the query request with the values of desired variables as the keywords to hydrology repository. In the context of this dissertation, the hydrology repository is the database that collects all the metadata for the distributed Web services discovered by the crawler proposed in Chapter 2.

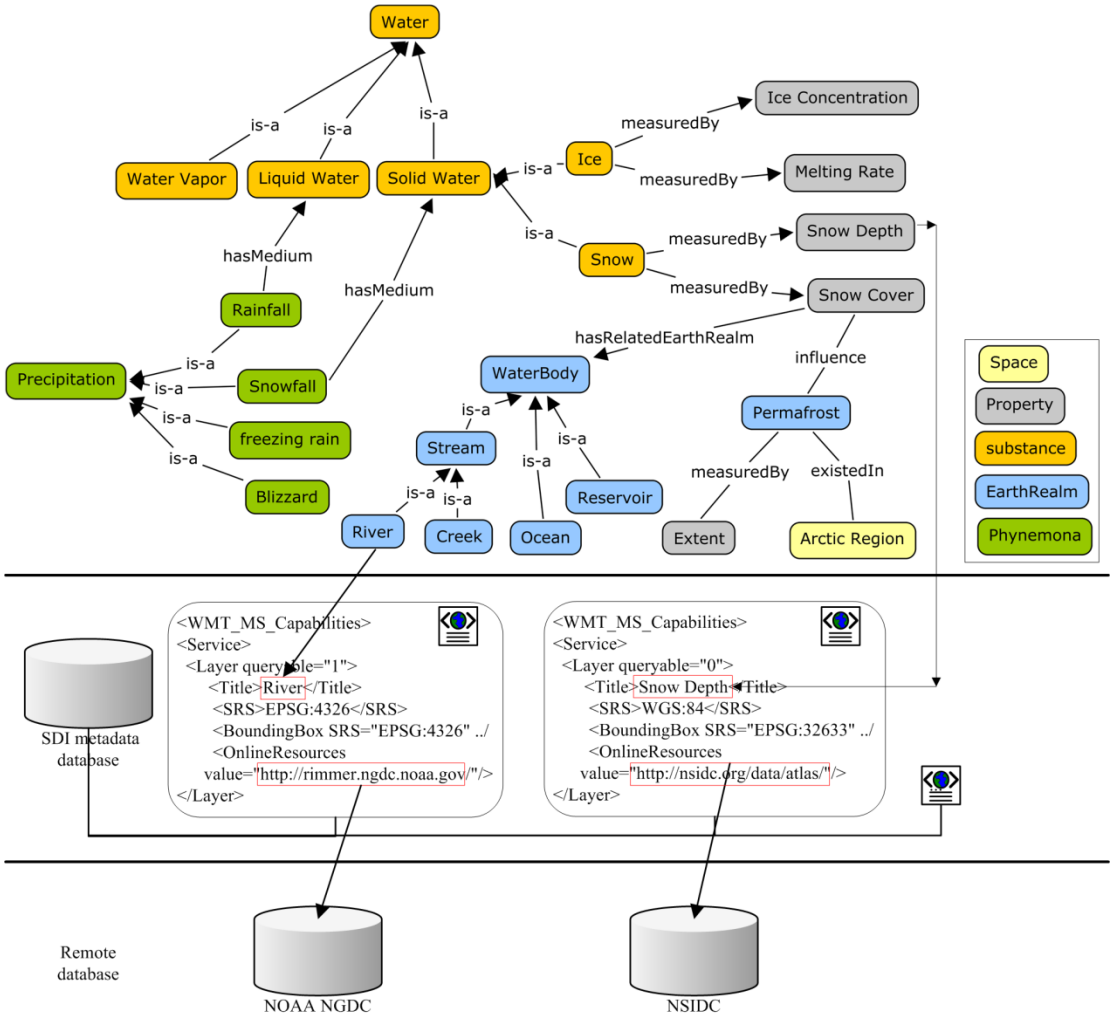


Figure 3.6 Ontology Fragment and its Linkage to Metadata and the Real Science Data.

Once the sub-queries in logical format are identified, they are translated into machine query language SPARQL (Protocol and RDF Query Language) for interacting with the KB. The following paragraphs show the SPARQL query for Q1a-Q1d.

SPARQL Query for Q1a:

```
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd:  <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
```

```
PREFIX owl:    <http://www.w3.org/2002/07/owl#>
PREFIX PhenomenaNS: <http://localhost/ontology/phenomena.owl#>
PREFIX PropertyNS: <http://localhost/ontology/property.owl#>
PREFIX SubstanceNS: <http://localhost/ontology/substance.owl#>
PREFIX EarthRealmNS: <http://localhost/ontology/earthrealm.owl#>
PREFIX ProcessNS: <http://localhost/ontology/process.owl#>

SELECT *
  WHERE {
?someSWClass rdfs:subClassOf SubstanceNS:SolidWater
}
```

SPARQL Query for Q1b:

```
PREFIX rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd:    <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl:    <http://www.w3.org/2002/07/owl#>
PREFIX PhenomenaNS: <http://localhost/ontology/phenomena.owl#>
PREFIX PropertyNS: <http://localhost/ontology/property.owl#>
PREFIX SubstanceNS: <http://localhost/ontology/substance.owl#>
PREFIX EarthRealmNS: <http://localhost/ontology/earthrealm.owl#>
PREFIX ProcessNS: <http://localhost/ontology/process.owl#>

SELECT *
WHERE {
someSWClass owl:onProperty ?p2 .
someSWClass rdf:type owl:Restriction .
someSWClass ?pre ?range .
FILTER (?pre != owl:onProperty)
FILTER (?pre != rdf:type)
FILTER (?pre != rdfs:subClassOf)
}
```

SPARQL Query for Q1c:

```
PREFIX rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd:    <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl:    <http://www.w3.org/2002/07/owl#>
```

```
PREFIX PhenomenaNS: <http://localhost/ontology/phenomena.owl#>
PREFIX PropertyNS: <http://localhost/ontology/property.owl#>
PREFIX SubstanceNS: <http://localhost/ontology/substance.owl#>
PREFIX EarthRealmNS: <http://localhost/ontology/earthrealm.owl#>
PREFIX ProcessNS: <http://localhost/ontology/process.owl#>

SELECT *
  WHERE {
?someStreamClass rdfs:subClassOf SubstanceNS:Stream
}
```

SPARQL Query for Q1d:

```
PREFIX rdf:    <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd:    <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl:    <http://www.w3.org/2002/07/owl#>
PREFIX PhenomenaNS: <http://localhost/ontology/phenomena.owl#>
PREFIX PropertyNS: <http://localhost/ontology/property.owl#>
PREFIX SubstanceNS: <http://localhost/ontology/substance.owl#>
PREFIX EarthRealmNS: <http://localhost/ontology/earthrealm.owl#>
PREFIX ProcessNS: <http://localhost/ontology/process.owl#>

SELECT *
  WHERE {
?Parameter PropertyNS:hasData ?data
?data PropertyNS:takePlaceIn ‘Arctic’^^xsd:String
?data PropertyNS:hasTime ‘Summer’^^xsd:String
}
```

These SPARQL queries are handled by an inference service. The inference service will first conduct non-predicate queries (such as Q1a and Q1c) and then predicate (such as Q1b) queries. This sequence is based on the following assumption: the associations (represented by predicate) with the parent class could be inherited by the child classes. Therefore, by retrieving more event-like classes by a non-predicate query and feeding

them into the predicate query, more candidate results will be retrieved. The inference engine adopted is the Jena tool introduced in Chapter 1. The processing procedures are: (1) the ontology is loaded into the memory or persistent storage maintained by Jena; (2) the DL-based queries are transformed into formal SPARQL [111] queries; (3) through the Jena query API, the sub queries are conducted and results for sub queries are retrieved; (4) query results are combined to obtain expanded and more specific information. The implicit association inference is enabled by recursively traversing the ontology tree to which the query term belongs. This relation is important because the associations of a class should contain both its own associations and its ascendants' associations.

After all the relevant keywords for the search are retrieved for semantic reasoning, they are directed to the metadata repository. Through matching the service metadata (middle layer in Figure 3.6) with given keywords, all relevant services that encapsulate the data are discovered. Following the metadata records, the URL that indicates the online resources of the actual scientific data is also found (lowermost layer in Figure 3.6). This way, not only are the data sources that are semantically related to their interests provided, but also the bridge between scientific data and scientific processes are well established.

3.5 Summary

In this chapter, an intelligent question answering methodology based on knowledge reasoning is introduced. The abstraction and modeling of domain knowledge, the way to construct a conceptual model, and the translation from conceptual model to machine understandable ontology are described. Meanwhile, a Web-based tool is developed to

support the collaborative ontology development for users with various backgrounds. Based on the populated hydrology ontology, a semantic reasoning algorithm is proposed to conduct syntax analysis, semantic analysis, semantic query decomposition and the correspondent data search automatically. The metadata repository used in this chapter is built from the Web services discovered by the Web crawler in Chapter 2. The semantic reasoning enables (1) query disambiguition by understanding a keyword with its exact meanings in the query context; (2) query-expansion with synonyms and related keywords discovered through knowledge reasoning.

Chapter 4: Semantic Similarity Determination: A Neural Net Methodology

4.1 Introduction

Semantic similarity is an important notion with two dimensions. It is used to describe the semantic distance between two concepts (either within a single ontology or among two different ontologies) or to measure the semantic distance between a concept and a word or a term. A "concept" is a class within an ontology; a "word" is a natural language element comprising information in a user's query or in a document on the WWW.

Sometimes, people use the terms "similar" and "related" interchangeably because they are both used to measure "relatedness." However, each focuses on different aspects of relatedness. For example, a car is more *related* to gasoline than to a bike, while a car is more *similar* to a bike than to gasoline. Identifying semantic relationships focuses on qualitatively measuring the structural relationships of concepts in an explicit hierarchy, such as Parent-Child, synonyms, etc. Identifying semantic similarity measures how closely two concepts are related by giving them a quantitative value.

Similarity measurement theories stem from psychological studies of the human ability to intuitively determine how similar two objects are and to quantify the similarity with a relation [112]. In the late 1980s, computer scientists in the field of AI engaged this research focusing on building computational models of ambiguous reasoning. With the

invention of the Semantic Web, researchers have attempted to combine similarity research with semantic technologies. Semantic similarity is central to many cognitive processes and plays an important role in the way humans process and reason about information [113]. It enables semantic interoperability between distributed information systems and web resources, thereby improving the quality of retrieval tasks for Internet users. Due to the growth of heterogeneous and independent data repositories, similarity-based information processing has become essential. It provides a measure of the degree of relatedness between concepts from different systems and domains [114]. Hence, the measurement of "semantic similarity" has emerged as an important topic in several areas of research.

A variety of applications are benefiting from similarity research, as large numbers of practical questions relate to disambiguation and distinction of concepts. For example, Google receives in total 240,000,000 questions asking about the difference from one concept to another and Microsoft Bing search receives in total 164,000,000 such questions. In hydrological science, semantic similarity is often used to identify objects that are conceptually close [115]. And according to Santos [116], an explicit model expressing a complete lattice would greatly help to eliminate the intrinsic vagueness and ambiguity of water features. In geospatial information science, the ontological modeling and similarity identification between geometric characteristics of self-objects and geographic relationships between spatial objects help improve the effectiveness of map generalization. In Web searches, traditional search engines are susceptible to the problems posed by the richness of natural language, especially the multitude of ways in

which the same concept can be expressed. Therefore, when making an effort to directly match user query terms with the database it is not possible to return satisfying results. Similarity measurement also provides a way to improve relevance ranking by eliminating conceptual ambiguities existing in user queries and metadata of documents [117].

4.2 A Use Case

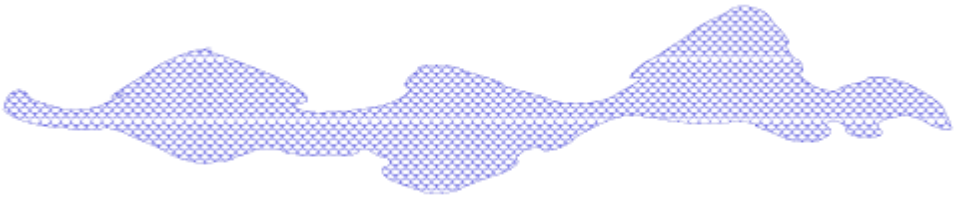


Figure 4.1 Vagueness in Water Features: Three Lakes or a Meandering River ? [117]

Conceptual vagueness is a ubiquitous problem in all Earth Science domains. Santos [117] demonstrated an example of vagueness in water features, as Figure 4.1 shows. This image may denote three lakes connected by channels or a meandering river. The decision from one interpretation to another depends on the existence of: (1) a complete set of information describing the above water feature, (2) a clear semantic definition of concepts "lake" and "river" to describe the boundaries of applicability of both terms, and (3) an effective algorithm that can precisely match the given feature to an existing concept by measuring the similarities based on the information provided in (1) and (2).

4.3 Previous Work

Many semantic similarity methods have been proposed in the past. In general, they can be categorized as edge-counting techniques [118][119], information theory based models [117][120], and feature matching models[115][121].

Edge-counting techniques are based on a network of semantic relations between concepts, and involve calculating the edge distances between objects in that network. A drawback of edge-counting is that it is hard to define the link distance in a uniform manner. In a practical KB, the distance between terminologies varies dramatically from categories and sub-categories, especially when some are much denser (have more subclasses) than others.

Information theory based models measure maximal information shared by two terminologies, calculated by the negative log likelihood of the concept. In this measurement, when probability increases, the informativeness decreases. So the higher the level a concept is, the higher the probability is, and thus the lower the information content it has. The statistics-based method lacks semantic support in the similarity measurement and therefore has a bias of human judgment.

In comparison to the above methods, the family of feature-based models, also called classic models, is the most prominent approach for similarity measurement. This approach is object-oriented, and describes resources by a set of features (such as components (roof and doors) and functionalities (residential or commercial use). The similarity between objects is a function of common and distinguishing features. For

example, Matching Distance Similarity Measure (MDSM) proposed by Rodrigues [115] is a feature-based model to measure the similarity between spatial entities. MDSM considers three kinds of features: functional feature, and features about attributes. The similarity calculation for each feature type is to count the common and differential features, and then apply them into Tversky's ratio model. The overall similarity is the linear sum of the weighted similarity values for each feature type.

Although prominent features are deterministic factors in human measurements of similarity, current feature-based models are still based on a KB with a simple logic. It is not suitable for mainstream knowledge representation, such as FOL and DL. The similarity equation in the MDSM model is a linear product of different feature sets. In contrast, humans' recognition of similarity is sometimes too complex to be simulated by these mathematical equations. We, of course, cannot rely on humans to provide the similarity for all the facts in the world since it would be too time-consuming and inflexible. Instead, there is a need of a "machine expert" to simulate the human perception process. The expert needs to be capable of answering "what if" questions efficiently. This requires the machine to have the ability to learn how to do tasks based on initial human experience.

4.4 Proposed Methodology

Artificial Neural Networks (ANN), with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. Another

desirable feature of neural nets is the ability to learn. This is undoubtedly the reason that several search engines (for example, MSN search) choose to utilize this model. With a neural net, an improved set of results could be produced over time to provide more relevant search results. The ANN model is inspired by biological neural networks, in which the human brain's information-processing capability is thought to emerge from a network of neurons. Since the 1940s, computer scientists have tried to model abstract "neurons" to capture the essence of neural computation in the brain. A neural network is strongly self-organized and can create its own representation of the information received during a learning period. Meanwhile, it is highly tolerant of noisy data and faults, which is very important to our application since human evaluation may have a big bias as well. Of the family of ANN algorithms, the *Multiple Layer Feed-Forward Neural Network* (MLFFN) is quite popular because of their ability to model complex relationships between output and input data. Adding more hidden units to the network makes it possible for MLFFN to represent any continuous, or even discontinuous, functions of the input parameters. This is a big advantage of MLFFN over other statistical algorithms proposed in the literature. In this dissertation, MLFFN is utilized to improve similarity measurement for semantic based ranking.

4.4.1 Problem Definition

Similarity measurement can be described as follows: from a collection of interrelated terminologies D, find a subset S of terminologies such that the similarity between each

element s_i of S and the given terminology t_g ranked by machine is highly correlated to human ranking. A mathematic description of this problem is as follows:

$$\text{For } \forall s_i,\ \Gamma(t_g, s_i) \rightarrow \mathrm{H}(t_g, s_i)$$

Where $i \trianglelefteq |S| \trianglelefteq |D|$, ($|X|$ equals the size of set X). H is a function for human ranking and Γ is the goal function that is simulated by proposed MLFFN based learning algorithm.

To expand this measurement to the scale of whole dataset D, $\Gamma(d_i, d_j)$ is calculated for $\forall d_i, d_j \in D,\ 0 \le i, j \trianglelefteq |D|$ and $i \neq j$, by which mean the distribution space for range of Γ could be obtained, as shown in Figure 4.2.

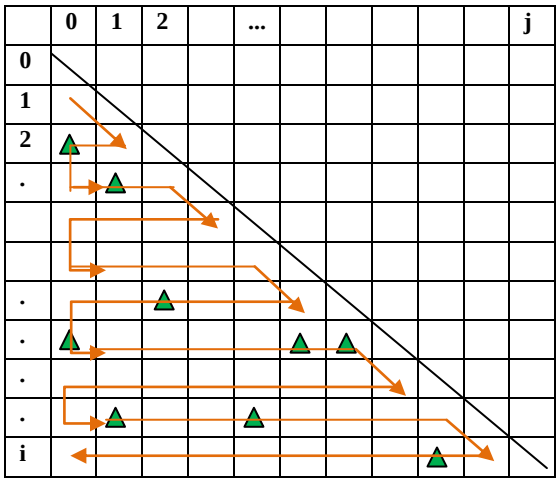


Figure 4.2 Distribution space for Γ. Lower triangle of the matrix filled with arrows demonstrates $\Gamma(d_i, d_j)$ and the green triangles in cells are the example training datasets available for the neural net.

4.4.2 MLFNN Algorithm

A MLFNN is used here to conduct numerical learning to simulate the knowledge propagation in a biological neural network. Figure 4.3 illustrates a general design of the multi-layer neural net which has multiple inputs and outputs:

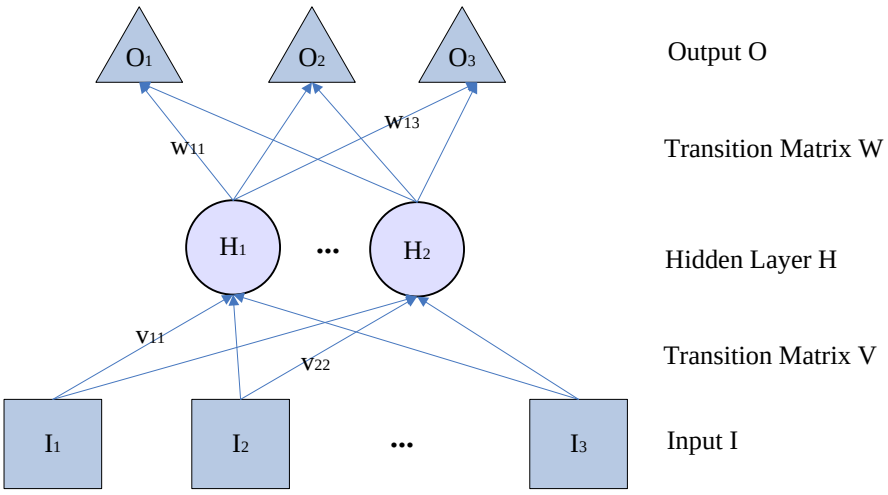


Figure 4.3 Design of a MLFNN.

The core of the algorithm is back propagation and forward propagation, where back propagation is used to train the neural network in order to get a stable transition matrix W and V, and forward propagation is used to measure the difference between predicted output and the desired output using current W and V. The error metric I propose to use here is mean squared error between output and desired correct output:

$$E = \frac{1}{n}\sum_{i=1}^{n}(C_i - O_i)^2$$

The detailed algorithm is:

1. Initialize W and V with given boundaries

2. Randomize given data D (a set of input vectors)

3. For each element in D,

   a. perform back propagation by:

$$\Delta W_{ij} = -\alpha \frac{\partial E}{\partial W_{ij}} = \alpha (C_i - O_i) O_i (1 - O_i) I_j$$

$$\Delta V_{jk} = \sum_i W_{ij} \Delta W_{ij} (1 - H_j) I_k$$

$$W = W + \Delta W$$

$$V = V + \Delta V$$

   b. perform forward propagation as follows:

$$H_j = \sigma(\sum_k V_{jk} I_k)$$
$$O = \sigma(\sum_j W_{ij} H_j)$$

   Where $\sigma(x) = \frac{1}{1+e^{-u}}$

   c. Calculate the mean squared error between each output and desired output, if the worst error is lower than a given good-minimum-error, then the network is finished training, returning V and W as two transition matrices. If the error is not lower than the given good-minimum-error, the algorithm will repeat back propagation to continue training the network.

4.4.3 The Acquisition of Prior Knowledge

Prior knowledge acts as the training dataset for the neural network. It determines how well the transition matrices could be built in the machine learning process. Although a neural net is highly tolerant of noisy data; completeness and representativeness of the prior knowledge is still of significant importance for the accuracy of semantic classification in similarity measurement. A neural net requires that the representation of knowledge be complete because the training process relies on the explicitly defined knowledge. Any uncertainty in the knowledge definition will lead to the failure of the predictive capability of neural net. Of all the existing machine language, *DL* is able to define a complete knowledge concentrating on a certain application area. It formalizes the knowledge structure by retaining an emphasis on definitions and properties of categories. DL is based on closed world assumption, by which those ground atomic sentences not asserted to be true are assumed to be false. Therefore, DL is suitable to represent knowledge and to build a domain KB. The knowledge structure and content defined in the KB should be representative and reflect the domain characteristics. As being discussed in Section 4.1, humans tend to measure similarity by comparing features of domain concepts, so like humans, there is a need to extract and provide the neural net a complete feature set that corresponds to the concepts defined in the KB. Sometimes only a few dominant features are deterministic to similarity measurement than the combination effects of all the features. Therefore, besides the completeness in the definition of feature types, the definition of prominent features of the concepts in a certain domain is also important.

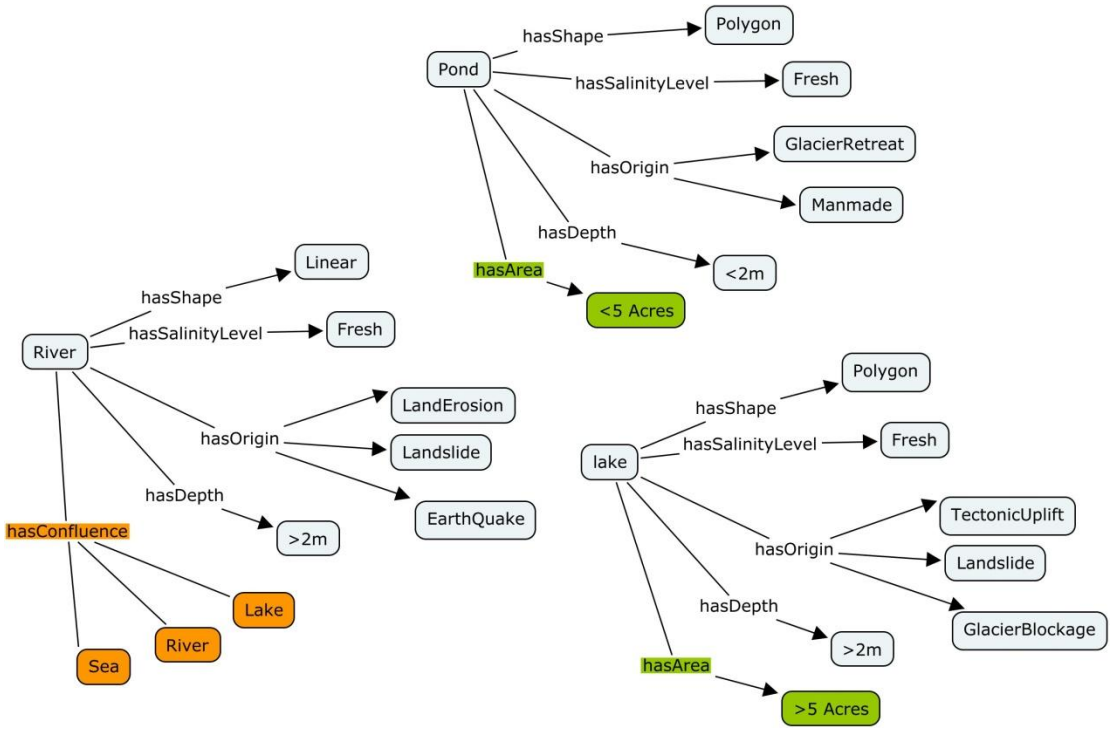


Figure 4.4 Semantic Definition of Three WaterBody Objects.

Figure 4.4 demonstrates semantic definitions of three WaterBody objects: river, lake and pond. All of them share the feature set {hasShape, hasSalinityLevel, hasOrigin, hasDepth}, although the range of the features are different. Feature nodes in orange are the ones only owned by "River;" nodes in green are the ones shared by “Pond” and “Lake” but “River”.

Comparing to MSDM and other feature-based models, which evaluate the similarity on the granularity for a set of features, the granularity in our measurement is on a single feature. Both feature and feature ranges in an assertion are considered. We believe it provides more accurate information in the similarity measurement than the existing models. An intuitive example is shown in Figure 4.4. Among the set of concepts {River,

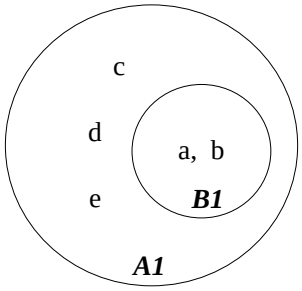
Lake, Pond} referring to a type of WaterBody, all have the same feature "hasShape." According to other feature based models, the contributions of the feature "hasShape" are the same for each concept pair. But practically, the contribution of the above feature is bigger in {Lake, Pond} pair than {Lake, River} and {Pond, River} because lake and pond both have an oval shape but river is always linear. Thus, by considering the value of each common feature, the proposed method can obtain more accurate values in similarity measurement than other feature matching models.

As a neural net is a numerically based learning algorithm, it requires all numerical input parameters. There is a need to calculate the contribution of each feature in comparison into numerical values. Three rules are declared for calculating the contribution:

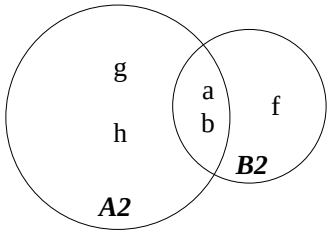
(1) For discrete features:

- **Rule I**

$$When\ |A_1 \cap B_1| = |A_2 \cap B_2| \neq \varnothing\ \ and\ \ |A_1 \cup B_1| = |A_2 \cup B_2| \neq \varnothing$$
$$Known\ that\ for\ feature\ f_i,\ A_k = \{x \mid x \in Range(t_{a_k}, f_i)\}, B_k = \{x \mid x \in Range(t_{b_k}, f_i)\}$$
$$Sim(f_i, t_{a_1}, t_{b_1}) \geq Sim(f_i, t_{a_2}, t_{b_2})$$
$$Therefore,\ if\ |A_1 - B_1| \cdot |B_1 - A_1| = 0$$
$$factor = threthhold \in [0.9, 1)$$
$$Sim(f_i, t_{a_1}, t_{b_1}) = \frac{|A_1 \cap B_1|}{|A_1 \cup B_1|};\ Sim(f_i, t_{a_2}, t_{b_2}) = factor \times \frac{|A_2 \cap B_2|}{|A_2 \cup B_2|}$$



A1={a,b,c,d,e}
B1={a,b}



A2={a,b,g,h}
B2={a,b,f}

- **Rule II**

$When\ A_1 \bigcap B_1 = \varnothing$

$CaseI\ : A_1 \bigcup B_1 \neq \varnothing$ and $|A_1| \cdot |B_1| = 0$

$CaseII\ : A_1 \bigcup B_1 = \varnothing$

$CaseIII : A_1 \bigcup B_1 \neq \varnothing$ and $|A_1| \cdot |B_1| \neq 0$

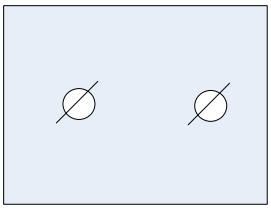
$Then : Sim(f_i, t_{a_1}, t_{b_1})_{caseIII} > Sim(f_i, t_{a_1}, t_{b_1})_{caseII} > Sim(f_i, t_{a_1}, t_{b_1})_{caseI}$

$Given : factor = \frac{1}{|A_1 \bigcup B_1|},\ factor2 = 0.1$

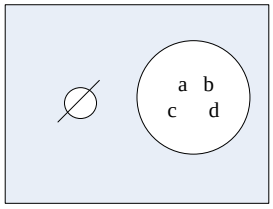
$Sim(f_i, t_{a_1}, t_{b_1})_{caseIII} = \frac{factor}{2}$

$Sim(f_i, t_{a_1}, t_{b_1})_{caseII} = \frac{factor2}{2}$

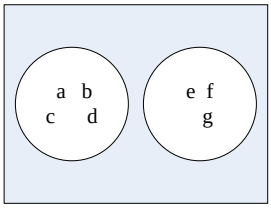
$Sim(f_i, t_{a_1}, t_{b_1})_{caseI} = 0$



Case I



Case II



Case III

(2) For continuous features

- **Rule III**

$if\ \ Range(t_{a_1}, f_i) = \{D_n\},\ A_1 = \{x,\ xx,\ xxx..\}\ \ |A_1| = n$

$Range(t_{b_1}, f_i) = \{D_m\},\ B_1 = \{x,\ xx,\ xxx,\ xxxx....\}\ \ |B_1| = m$

x is a unit element that associates to feature f_i, then apply the calcuation listed in Rule II

The contribution of feature f_i in measuring the similarity between objects t_{a_1} and t_{b_1} is the ratio between shared members of A_1 and B_1, and the range they cover in total. So the more of the same member shared by A_1 and B_1, the more contribution the feature makes to the similarity between the two objects. $factor$ and $factor2$ in Rule II are tuning parameters, distinguishing the cases of (1) the two objects not sharing the feature f_i (Case I and Case II); (2) two objects sharing the same feature, but not having any intersection in the range set (Case III). For example, both WaterBody "Lake" and "River" have common feature "hasOrigin." The range of this feature for "Lake" is {TectonicUplift, Landslide, GlacierBlockage}, and that for "River" is {LandErosion, Landslide, Earthquake}. According to Rule II, the contribution of feature "hasOrigin" to the similarity of {lake, river} is 0.167 rather than 0.

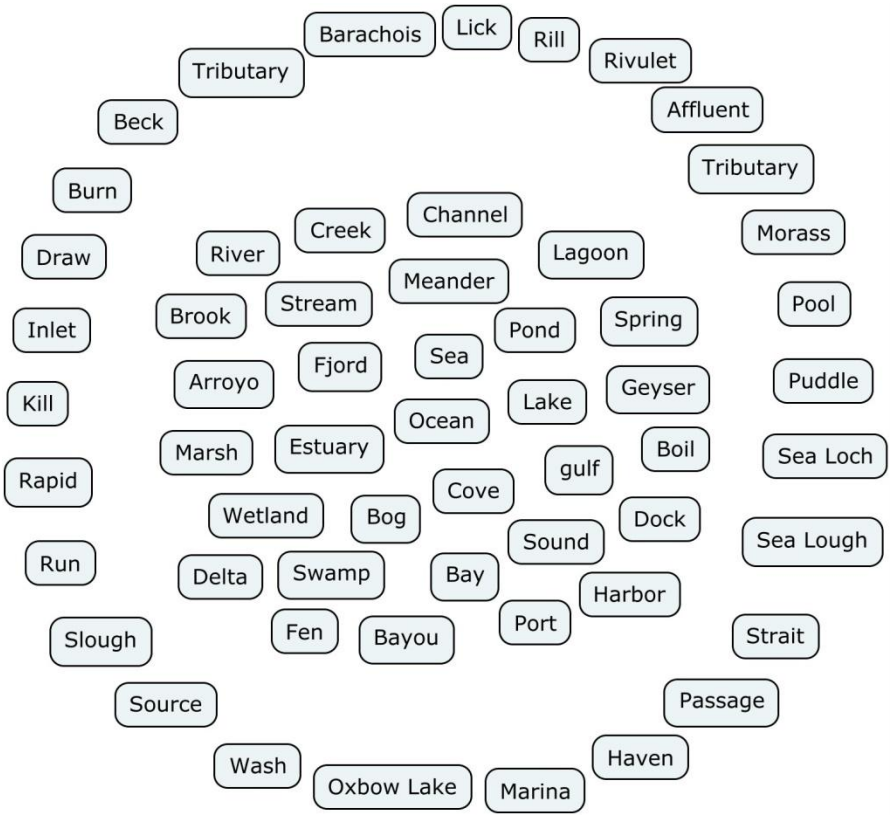


Figure 4.5 Core WaterBody Objects Used for Training.

Once rules for calculating contributions of both common and differential features are defined, the input pattern of the neural net is mapped from available data. Figure 4.5 shows the core WaterBody objects used as the training set. The neural net input includes a vector of multi-dimensional parameters and a known output result. Features are mapped onto the multi-dimensional parameters and the value of each parameter is the contribution of the specific feature to the similarity of the two objects in pair. The known output result is obtained from human ranking results on sample data.

Training process and workflow:

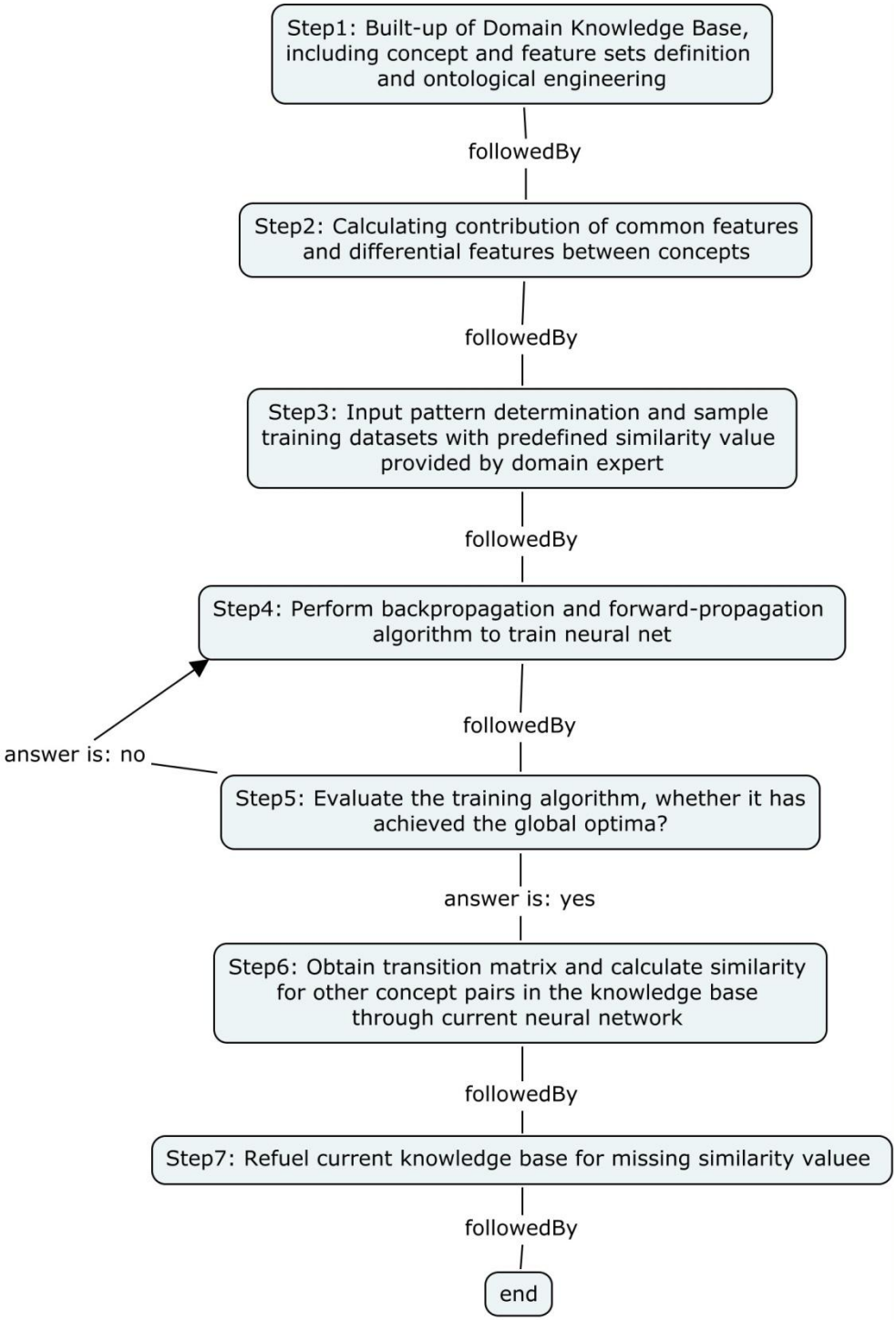


Figure 4.6 Training Process and Workflow.

Through the whole training process, the goal function $\Gamma(d_i, d_j)$ introduced in Section 4.2 can be obtained in the formats of transition matrices.

Another issue for the similarity measurement is the timely update of a similarity matrix as the size of KB increases. Once a new instance is populated into the KB, its similarity with other instances in the KB will be calculated automatically. Using the obtained transition matrix, a forward propagation can be conducted N (number of instances in KB) times to calculate missing similarity values. This achievement is based on the premise that the schema (object-level) of the KB is consistent, or the whole training process needs to be conducted again for new transition matrices.

4.5 Assessing the ANN-based Similarity Measure Approach

A commonly used experiment for assessing the ANN-based similarity measure approach is to distribute to 38 undergraduate subjects 30 pairs of nouns that covered high, intermediate, and low levels of similarity [122]. In this dissertation, the design of experiments was slightly different than the one used in [122] because: (1) the concepts measured are specialized for the hydrology domain, so measurements obtained from subjects who have little background in this domain will be biased; (2) as a machine learning algorithm, ANN requires a large number of sample data to train and validate the network.

Therefore, a new experiment was designed to satisfy the criteria mentioned above. The human subjects were asked to provide similarity scores for three groups of concept pairs. Subjects ranked the concept pairs from least to most similar, the lowest score was 0

for the least similar pair in a group while score 100 was for the most similar pair. When scoring the similarity of one pair, the subject had to consider the relative distance of the similarity of this pair to that of other pairs in the same group. The three groups are linear WaterBody, non-linear open WaterBody and non-linear WaterBody (wetlands). According to the background of the human subjects (graduate students in Earth Science or hydrology experts), different surveys were given separately. The survey for graduate subjects included 10 pairs of terms in each group. The survey designed for hydrology experts included all questions in the survey designed for graduate subjects, plus 33 other pairs. The extra pairs contained concepts from across groups, e.g. one is from the linear, while the other is from the non-linear WaterBody group, e.g. (River, Lake), as shown in Table 4.1.

Table 4.1 Survey Conducted to Human Subjects.

Subject Type	Survey A (Linear)	Survey B (Non-linear I)	Survey C (Non-linear II)	Survey D (Cross-group)
Both	%(River,Fjord) %(River,Creek) %(River,Brook) %(River,Bayou) %(Creek,Fjord) %(Creek,Arroyo) %(Creek, Brook) %(Brook,Arroyo) %(Creek, Bayou) %(Bayou,Brook)	%(Sea,Ocean) %(Sea,Bay) %(Sea,Gulf) %(Sea,Cove) %(Sea,Harbor) %(Sea,Port) %(Sea,Dock) %(Bay,Gulf) %(Bay,Cove) %(Harbor,Port) %(Harbor,Dock) %(Port,Dock)	%(Swamp,Marsh) %(Wetland,Swamp) %(Wetland,Marsh) %(Wetland,Bog) %(Wetland,Fen) %(Swamp,Bog) %(Swamp,Fen) %(Bog,Fen)	
Expert subject only	%(Bayou,Arroyo) %(Bayou,Fjord) %(Brook,Fjord)	%(Ocean,Bay) %(Ocean,Gulf) %(Ocean,Cove)	%(Wetland,Fen) %(Marsh,Bog) %(Marsh,Fen)	%(Lake,Arroyo) %(Lake,Bayou)

	%(Fjord,Arroyo) %(River,Arroyo)	%(Ocean,Harbor) %(Ocean,Port) %(Ocean,Dock) %(Bay,Harbor) %(Bay,Port) %(Bay,Dock) %(Gulf,Cove) %(Gulf,Harbor) %(Gulf,Port) %(Gulf,Dock) %(Cove,Harbor) %(Cove,Port) %(Cove,Dock)		%(Lake,Brook) %(Lake,Creek) %(Lake,Fjord) %(Lake,River) %(Pond,Arroyo) %(Pond,Bayou) %(Pond,Brook) %(Pond,Creek) %(Pond,Fjord) %(Pond,Lake) %(Pond,River)

Based on the collected experimental data, the following assessment was conducted to evaluate the performance of ANN when enabling the automated similarity measurement as described in the following sections. Section 4.5.1 describes how quickly the ANN can converge when the learning rate of the neural net is set differently. Section 4.5.2 describes how the number of hidden neurons influences the performance of the neural network. Section 4.5.3 describes the analysis of the robustness of the designed ANN in terms of decreasing the number of sample data and the difference in recognition of spatial concept between subject Type A (graduate student) and subject Type B (hydrology expert).

4.5.1 Quickness of Convergence v.s. Learning Rate

The learning rate controls the speed of ANN learning by affecting the changes being made to the weights of transition matrices at each step. The performance of the ANN algorithm is very sensitive to the proper setting of the learning rate [123]. If the changes

applied to the weights are too small, the algorithm will take too long to converge. If the changes are too large, the algorithm becomes unstable and bounds around the error surface. This experiment determined the optimum network for automated similarity measurement through the result from this learning rate investigation. Here, "optimum" is measured by the Mean Square Error (MSE, discussed in Section 4.4.2) between the network outputs and the target outputs obtained from the human-subject experiments. The initial parameters used for training the network are shown in Table 4.2. Parameter 1 is the largest number of steps that the ANN is going to run; Parameter 2 is measured by MSE, 10^-3 means the ANN will stop training if MSE<10^-3; Parameter 3 is the initial learning rate, in this experiment the learning rate was set to different numbers in each training process; Parameter 4 set the training expiration time to infinity. The introduction of Parameter 5 cut down the learning time and efficiently prevented the network from sticking at local optima.

Table 4.2 Training parameters.

No.	Parameter	Value
1	Number of Epoches	2000 5000 8000
2	Goal of performance function	10^-3
3	Initial Learning Rate	0.1
4	Training Time	Inf.
5	momentum coefficient	0.9

Figure 4.7 shows the neural network learning rate experimental results by the number of epochs. The X axis indicates learning rate ranging from 0.1 to 0.9 with interval 0.1, while the Y axis indicates how many times the ANN must be trained until the result converges. As the network training uses heuristic techniques, it tends to become trapped

in the local optimum due to the nature of the gradient descent algorithm from which these heuristic techniques were developed [124]. The strategy used here to compensate for the sticking problem was to retrain the network until the result achieved the performance function goal (MSE<10^-3). The Y axis records this number.

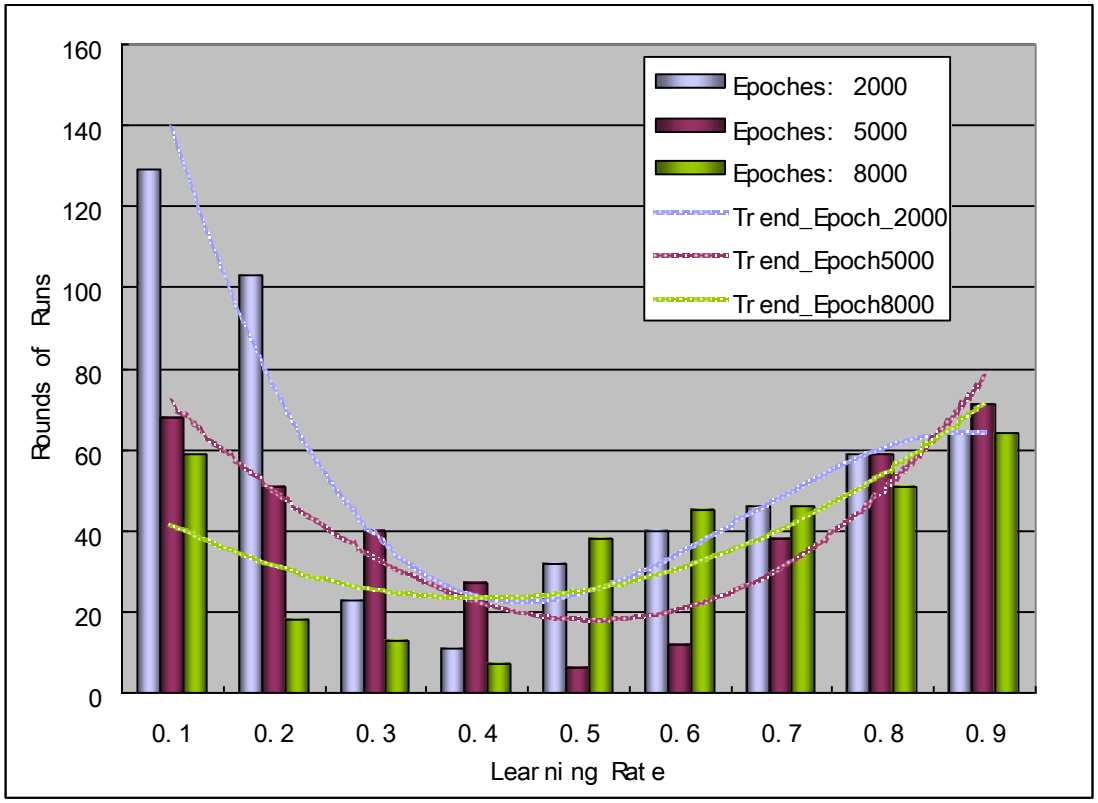


Figure 4.7 Number of Training Runs for the ANN Needed in Terms of Various Learning Rates.

It is not hard to tell that when epoch is set to be large, it is more likely to make MSE between training output and target output within the tolerable range; therefore, the network needs fewer training runs. But the difference in complexity levels of different problems determines that the above assertion is not necessarily true. For the automated

similarity measurement problem, the assertion is true only when the learning rate is less than 0.3. When the learning rate is more than 0.5, the setting of the epoch will not influence the number of training runs required. Another observation is that when the learning rate is less than 0.4, the number of training runs for epoch = 2000 decreases much faster than the decreasing rate of training runs for epoch = 5000 and epoch =8000. It means that the designed neural network is most sensitive to change in learning rate when epoch is set to 2000. Meanwhile, the trend curves in Figure 4.7 show that for the same epoch of each training process, the necessary training runs decrease when the learning rate increases until the learning rate reaches 0.4. Based on the above analysis, when the epoch equals 2000 and the learning rate equals 0.4, the ANN performs the best and therefore, these as parameters were chosen for the following experiments.

4.5.2 Prediction Accuracy v.s. Number of Hidden Nodes

One great advantage of the ANN model is its ability to predict. Once experimental data are collected from human subjects, the neural network can be well trained. Using the trained network, the ANN model can provide automatic ranking for the pairs of concepts that are not ranked by humans. In order to accomplish this capacity, the experimental results from the human subjects were divided into two sets: 90% of the results are considered as the testing set and the remaining 10% were considered as the validation set.

The correlation between the computational similarity models and human's judgment have been widely used in previous studies to measure the accuracy of the model [117][125]. A correlation of 0.6 using a semantic-distance approach, 0.79 using an

information-content approach, and 0.83 using an extended-distance approach are reported in the literature [117]. In this dissertation, a Spearman rank correlation coefficient r is used as one factor to investigate the association between the results from the trained ANN model and validation sets from human subjects.

$$r(X,Y) = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2 \sum_i (y_i - \bar{y})^2}},$$

where X is the set of predicted similarities obtained from well-trained network and Y is the similarity values ranked by subjects.

The larger r is, the more accurate that the ANN model is in predicting the similarity. The coefficient r only provides relevant qualified measurement of correlation for the two sets of data, the ANN generated set and the validation set. A higher correlation coefficient between the above datasets does not mean that the values in each corresponding pair are closer. A more accurate factor to measure the "prediction error" is the Square root of MSE (SMSE) between values of each pair ranked by subjects and predicted by the ANN model. SMSE can be represented as:

$$SMSE = \sqrt{\frac{\sum_{i=1}^{n} (ANNp_i - Hp_i)^2}{n}}$$

Therefore, the goal of an ANN model is to both maximize the coefficient r and minimize the SMSE. Figure 4.8 shows both Spearman coefficients (value in %) and the SMSE values. We can tell that from the nine hidden node settings of the ANN model,

five of them result in high correlation (>85%) when making predictions. This means that the proposed ANN model is reliable in making predictions and the high correlation shows that the ANN approach is better than most of the models proposed before. The best performance (r=94.86%, SMSE=11.47) for the trained neural network occurs when hidden neuron equals 9.

As the number of hidden neuron determines the complexity of the neural network, although the ANN with different neuron settings all satisfy the goal (MSE<1) when training the network, ANN will still cause overfitting or underfitting problems with too much or too few hidden neurons. In Figure 4.8, we can tell that when hidden number is 3 or 4, the network is not sufficiently complex and fails to detect fully the signal in the sample dataset. Therefore, this model leads to underfitting with low correlation and high SMSE in the prediction. When the hidden neuron is equal to 10 or 11, the performance of the ANN declined, probably because the network is overly complex, which leads to an overfitting problem where the noise is well fitted and consequently makes the prediction not close enough to the range of the training data.

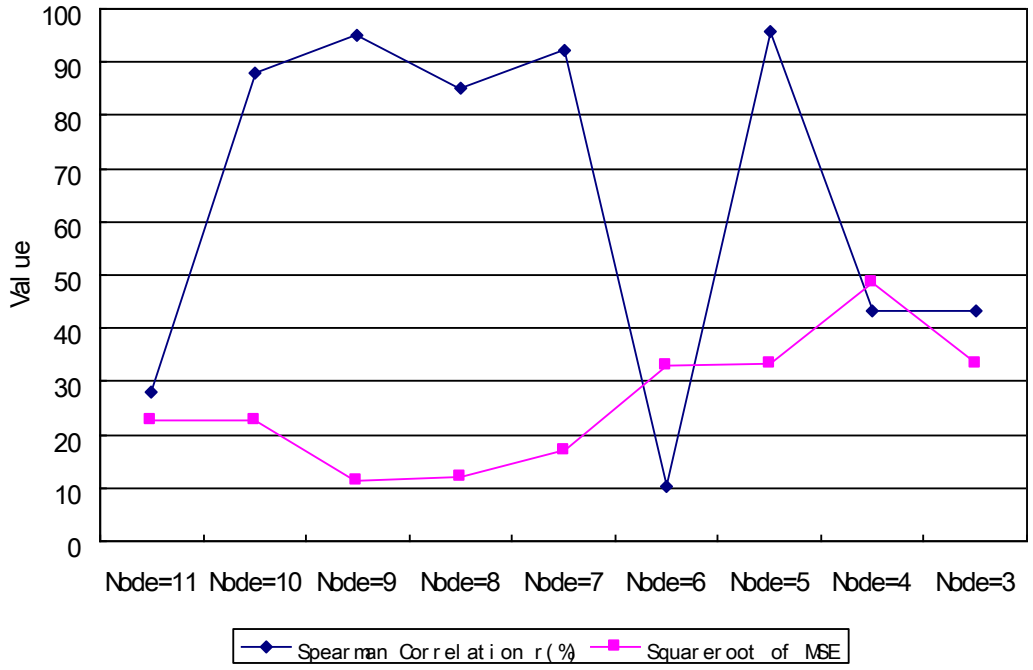


Figure 4.8 Prediction Accuracy by Different Number of Hidden Neurons.

4.5.3 Accuracy of ANN Prediction v.s. Background of the Subjects

This experiment examines the accuracy of ANN prediction given the different backgrounds of the human subjects whose responses were the basis for the training and validation datasets. The ANN was trained with the optimal learning rate (0.4) and the optimal hidden neurons (9) obtained from the above experiments. As the sample dataset from human subjects for each group of pairs of WaterBody objects is relatively small (refer to the number of pairs in Table 4.1), ranking data from other groups are borrowed to make sure that the total number of samples for the ANN training group was equal to or more than the number of total features (Num(feature)=17) of the WaterBody objects. 3 pairs of the WaterBody objects in each group were used for validation and the rest of the pairs were used for training. As Table 4.3 shows, the correlation coefficients acquired for

both types of the subjects are all above 70% and still lower than the number acquired in Section 4.5.2 which conducted on a larger size of sample data. The ANN trained by data collected from graduate subjects has a lower correlation coefficient than that collected from the expert subjects, meaning that more noisy data exists in the survey of graduate subjects. Tracking their similarity rankings, suggests two primary reasons for the increased noise: (a) graduate subjects tend to rank the similarity based more on the familiarity of the spatial concepts rather than the actual meaning; for example, "Bog" and "Fen" have many common features including shape, size, how they are developed, water source and water salinity. But most graduate subjects gave these pairs a low rank due to unfamiliarity with these WaterBody terms. (b) Misunderstanding of the spatial concepts leads to much bias of the ranking results. In comparison, ranking results from hydrology experts is more reliable. From this experiment, we can conclude that collecting enough (>3 times of total feature sets in our case) reliable sample data is important for the ANN model to perform accurate prediction.

Table 4.3 Accuracy of the ANN Prediction for Graduate Subjects and Expert Subjects.

Correlation / Subject Type	Linear	Non-linear	
		Non-linear I	Non-linear II
Graduate Subject	0.71	0.82	0.79
Expert Subject	0.81	0.85	0.91

4.6 Summary

This chapter introduces a novel feature-based approach to utilize ANN to best simulate human's similarity ranking process based on training artificial neurons with

sample data collected from human subjects. The collection and ontological modeling of spatial objects, the calculation of contribution for each feature of any two spatial objects and the ANN design are introduced. In several different experiments, the ANN-based approach achieved good performance in terms of both correlation and mean standard error when predicting the association between ANN prediction and human subjects' ranking.

Chapter 5 Applications

This chapter introduces two research projects: the USGS Arctic Spatial Data Infrastructure and the ESIP Semantic Web Testbed in which the proposed approaches in Chapter 2, 3 and 4 are applied to solve real world problems.

5.1 Arctic Spatial Data Infrastructure

The Arctic region has experienced the greatest climate change impact over the past century. The change has increased significantly over the last few decades [126]. For example, in the past twenty years, the melting rate of sea ice in the Arctic Ocean has increased rapidly [127]. If this trend continues, the Arctic Ocean will be completely ice-free during the summer somewhere between 2013 and 2030 [128]. The loss of solid water resources leads directly to the sea level rising, endangering the habitats of polar life. Accurate monitoring of the large-scale dimensions of solid water concentrations in the high latitudes of the Northern Hemisphere is a crucial task for Arctic scientists.

To accomplish this task, it is necessary to integrate disparate and distributed hydrology data. Scientists face technical challenges when implementing an interoperable cyberinfrastructure to facilitate the discovery, federation and seamless fusion of scientific data from disparate and distributed resources. A Spatial Data infrastructure (SDI), aiming to address this challenge, is fundamentally about facilitation and coordination of the

exchange and sharing of geospatial data between stakeholders in the spatial data community [129][130]. SDI research encompasses the policies, data, technologies, standards, delivery mechanisms and financial and human resources necessary to ensure the availability and accessibility of the spatial data [131]. Over the past decade, many SDIs have been built at a national or regional scale [132]-[137]. However, very few SDIs were specifically developed to support Arctic research. Several key issues were addressed by the methodologies proposed in this dissertation: (1) the active crawler enables the automatic discovery of spatial data that exists in a distributed computing environment. These collected services are stored in the metadata repository and are publically available to researchers through the Internet. (2) The build-up of a hydrology ontology models the domain knowledge in a machine understandable way by explicitly defining the conceptualization and the relationships; (3) the reasoning algorithm provides an intelligent resource search mechanism to support real-time decision making for Arctic studies.

With the support of the above technologies, an integrated science analysis environment can be enabled. For example, to study the influence of melting snow and sea ice on habitat changes of polar wildlife, a scientist 'enters' the ASDI and the following scenario unfolds:

(1) Initially, an automated service in the SDI discovery identifies all the distributed Arctic data resources across a public network and places them in a data repository. The scientist, who is the data consumer, is only presented with a transparent search

interface, which hides all implementation details.

(2) The scientist opens the user interface of the Arctic SDI and assembles the query elements required to narrow down the available information. The query elements should be geophysical parameters, such as snow cover, sea ice concentration, precipitation and biodiversity.

(3) The search request is handled by the "Search Service" and passed to the virtual repository. Through semantic reasoning, all the relevant datasets are discovered. Results from disparate sources are collected and returned to the scientist. The results are then visualized in a 2D or 3D format.

(4) Examining the results composited, the scientist decides whether to acquire more datasets for further analysis. By clicking the embedded URLs, the scientist gains direct access to the needed resources to feed a simulation model or conduct a cross-correlation of multiple variables. For example, when a researcher explores "the influence of solid water dynamics in the Arctic region to bio-habitat," he can start the query by typing in a most intuitive keyword. The semantic search service (Figure 5.1, Box 1) will generate a chaining workflow to identify all relevant datasets (Figure 5.1, Box 2) after the spatial and temporal subset (Figure 5.1, Box 4). As shown in Figure 5.1, once a query term "snow" is given, knowledge reasoning could infer "Snowfall" as a synonym; "Precipitation" as a broader term, "Water" as a related substance, "Precipitation," "Cloud," and "Wind" as related Phenomena, "Pressure" as a related Property and "Deposition" as a related Process. In addition, "Rivers," "Bio_Sample," and "Ice" are automatically inferred for service composition. "Blue Marble" is set as

the base map. After semantic reasoning, scientists obtain a rich dictionary of candidate resources to choose the best resources they need. The evaluation of the 'best' is based on the quality of the discovered data (e.g response time) and the quality of information is indicated by an icon bar displayed on the results panel (Box 2). After selecting the most suitable service by the client, an integration service is invoked to automatically overlay datasets and display them in a 2-D or 3-D client (Figure 5.1, Box 3; [138]). The produced imagery in the map client indicates a phenomenon that the bio-habitats are mostly distributed along the coast and within large water bodies. To identify the science principle, river data, snow cover and ice extent can be used. The scientist can also generate a time-series based animation (Figure 5.1, Box 5) to discover how the variation of solid water concentration influences the immigration of bio-habitats.

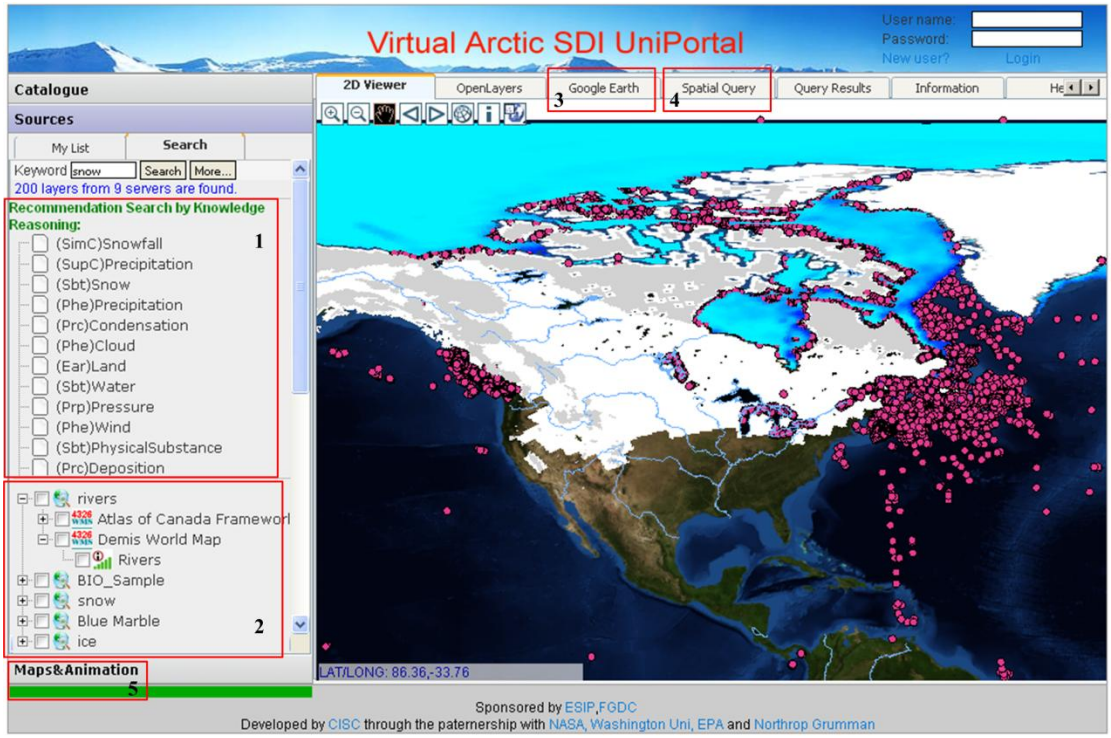


Figure 5.1 Prototype of the Arctic Spatial Data Infrastructure. The bold numbers (1-4) denote the corresponding boxes mentioned in the main text.

5.2 ESIP Semantic Web Testbed

Semantic Web technology has been driving the next generation of the Web where the focus is on the role of semantics for automated approaches to exploit web resources [139]. This technology was first proposed by Tim Berners Lee [25], and has become a critical research topic in AI and related fields. The core of semantic Web research involves two well-recognized critical enabling capabilities: (1) ontology generation [140][141] and (2) automated resource annotation, reasoning and integration [142][143][144]. The introduction of the Semantic Web concept inspires domain experts. Scientists from medical domain [145], biology [146], chemistry [147], etc. rely on this

technique to solve problems in various domains. In the Earth Science domain, some uses of reasoning engines, such as Pellet and Neosis, are limited to query search expansion. The ESIP semantic search testbed [148]-[149] aims to completely utilize the full knowledge encoded in the ontologies and provides a better search experience to end users. In this project, the ontology is built up through the guidelines proposed in Chapter 3. A reasoning engine that implements the algorithm proposed in this dissertation enables automatic assembly of shared services. It is also able to answer questions on the fly to support better decision making.

Figure 5.2 shows the architecture for the ESIP Semantic Web Testbed. It is composed of three major elements: registration, semantic search and ontology management. For semantic registration, a Web-based ontology editor – Web Protégé (Figure 5.3) and RDF direct registration GUI (Figure 5.4) are provided. The registered knowledge is managed in the ontology repository using an open source framework Sesame 2.3 and MySQL 5.0. The inference services that implement the semantic reasoning algorithm are deployed using Apache Tomcat 5.5.28 and Apache Axis 1.2. These services are invoked by the semantic search client (Figure 5.5) using the Asynchronous JavaScript and XML (AJAX) technology to reduce the processing load on the client side and to improve efficiency. The data sources that the semantic search client links to are popular geospatial web catalogues, including GOS (Geospatial One Stop), GCMD (Global Climate Master Directory), NCDC (National Climatic Data Center) and ECHO (Earth Observation ClearingHouse). Services discovered by the crawler are registered into GOS portal and are made available for the semantic search client. The

knowledge that is inferred (Box 2, Figure 5.5) is organized and displayed in a tree-based style and is visualized using an open source tool Prefuse (Box 4, Figure 5.5). The similarity matrix obtained from the neural network training described in Chapter 4 was integrated to rank the relevance of recommended terms to the input keyword (Box 2, Figure 5.5). An ontology expert can edit the inferred knowledge directly if he finds they are not complete or accurate (Box 3, Figure 5.5). The changes are reflected immediately to any client accessing to the ontology repository.

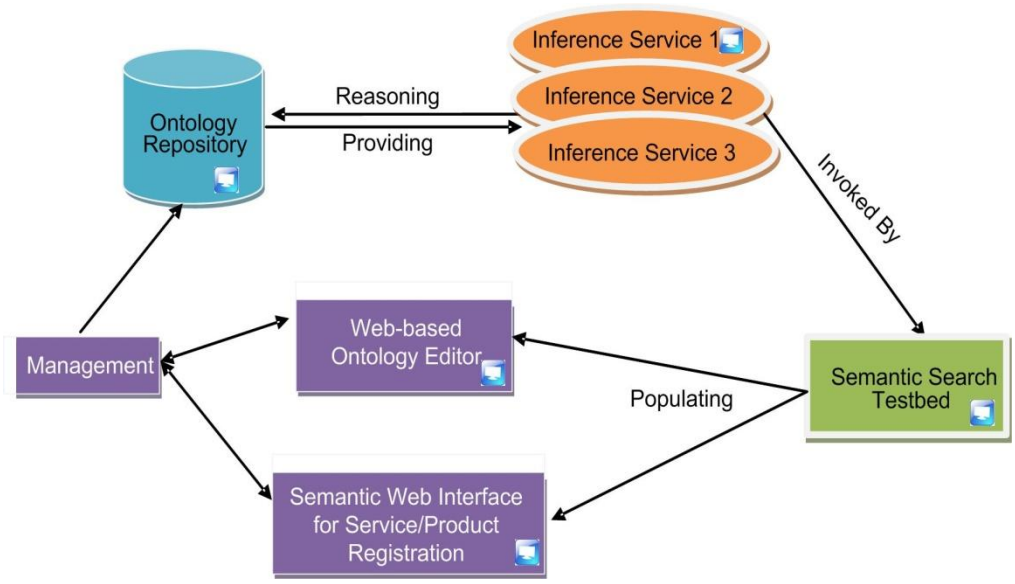


Figure 5.2 Architecture of the ESIP Semantic Web Testbed

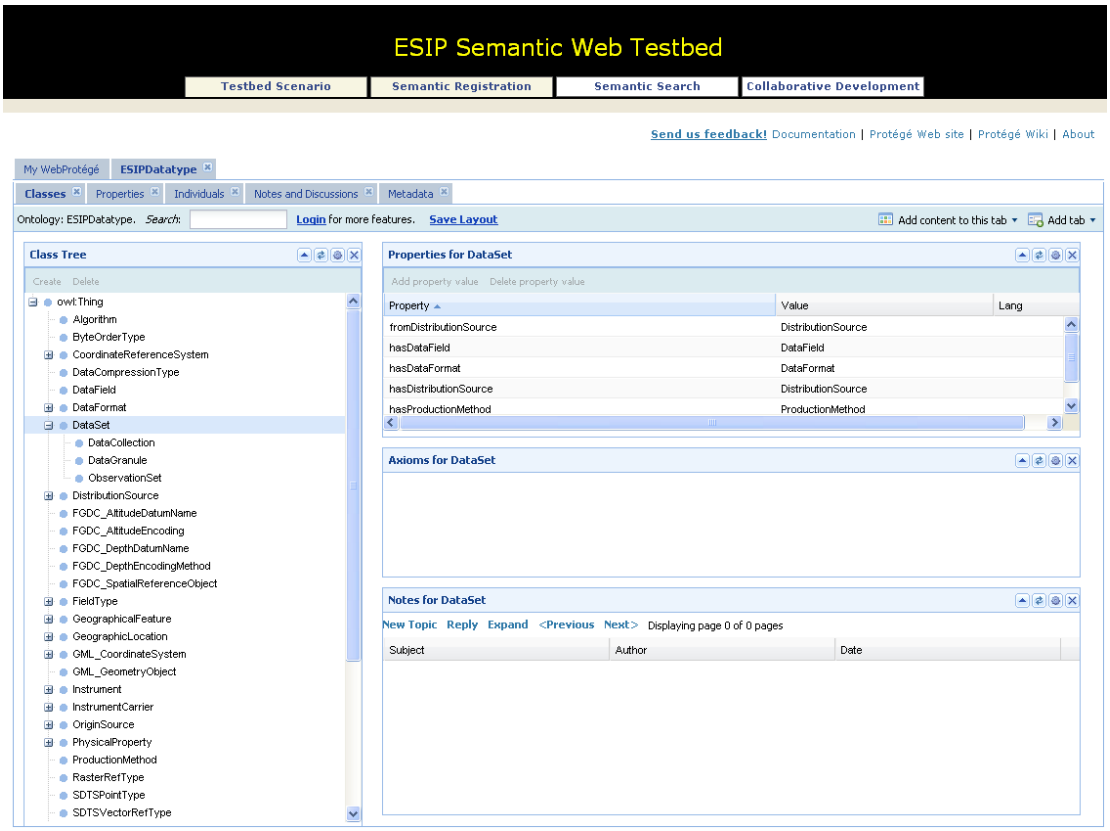


Figure 5.3 A Web-protégé based GUI for Semantic Registration.

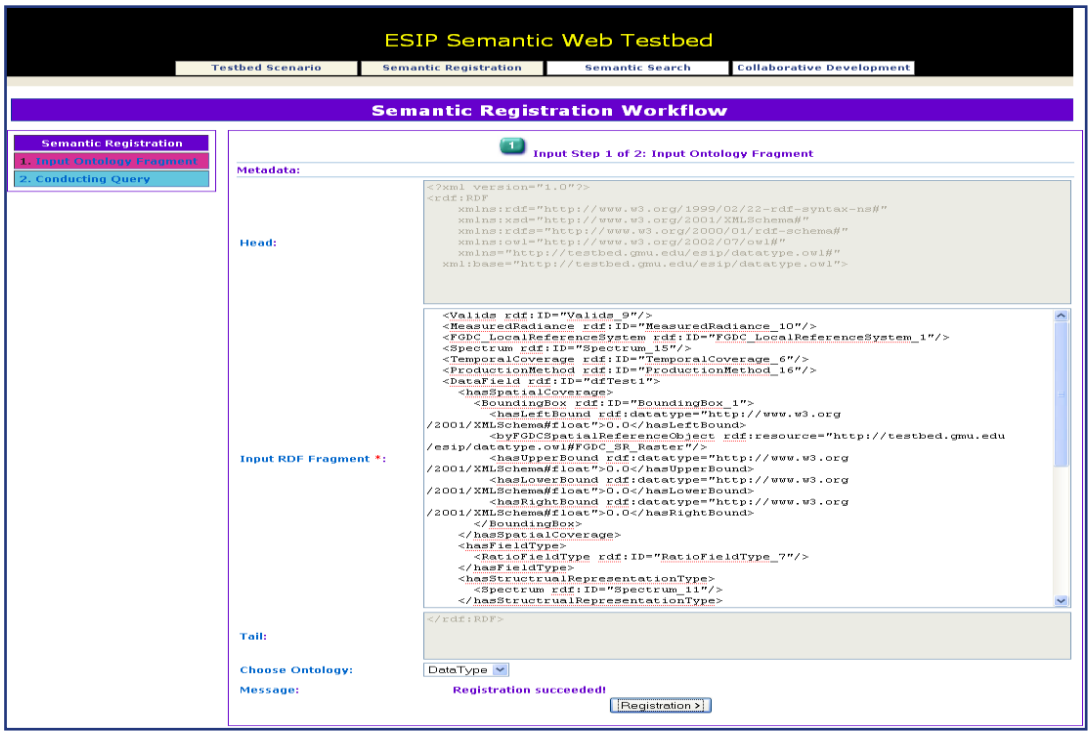


Figure 5.4 Direct RDF Fragment Registration Interface.

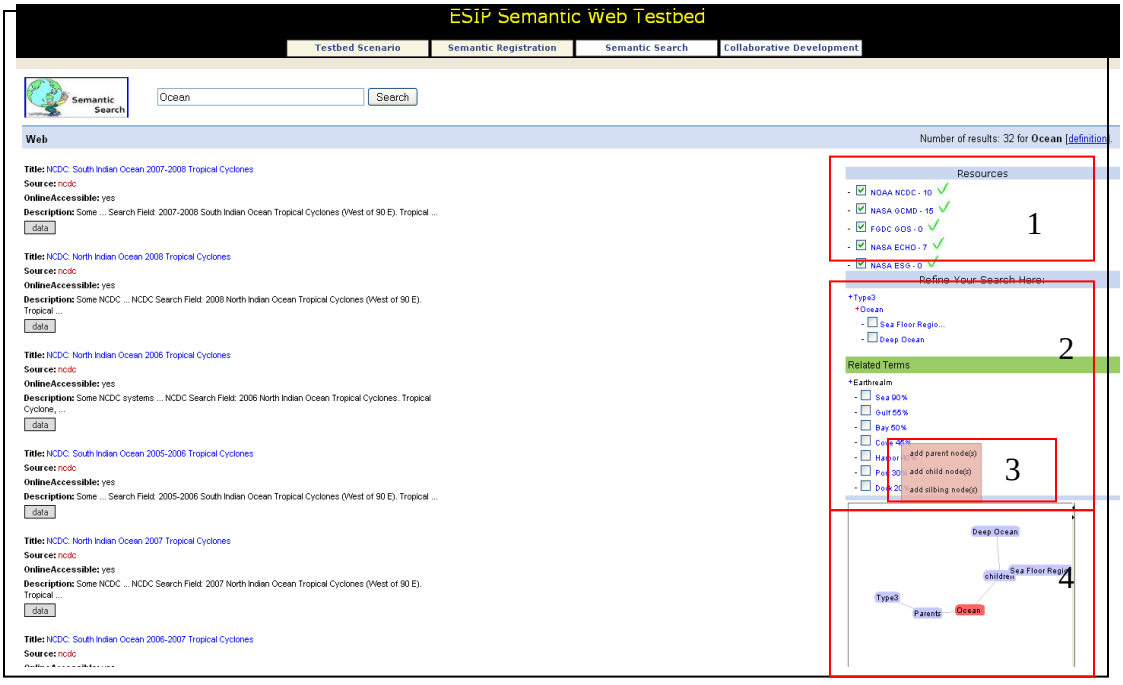


Figure 5.5 Semantic Search and Visualization Client.

Chapter 6 Conclusions and Future Research

6.1 Conclusions

This dissertation presents a number of techniques that address a fundamental question in the problem of information discovery: how to discover automatically huge amounts of geospatial data dispersed widely on the Web? Once this information is found, how can this information be encoded from human-readable format to machine understandable format? How to make the machine incorporate human intelligence to answer various search questions? Or more intuitively, how to make the right connection between the human and the resources he/she needs?

Developing a search engine that can address the above questions is extremely important when conducting scientific analysis or when supporting critical real-time decision-making. This dissertation discussed three essential parts (data discovery, semantic reasoning and semantic similarity measurement) in designing a geospatial search engine for intelligent hydrology data discovery.

Data discovery involves the collection of as many distributed resources as possible on the Internet. The set of the discovered data serves as a starting point from which a

more extensive search can be conducted. The more resources found and put into the repository, the better chance that a user can acquire the most relevant data for their purposes. The proposed active web crawler facilitates the automatic discovery of online geospatial resources.

The goal of semantic reasoning is to understand search requests and provide suitable data candidates that exist in the database collected by the Web crawler. This dissertation proposed a principle for modeling the domain knowledge, developed a Web-based tool to support collaborative ontology development, and employed semantic technology to realize intelligent question answering.

Semantic similarity measurement uncovers the hidden pattern residing in the KB and simulates how humans perceive similarity by deploying a neural network based approach. Given that semantic reasoning helps to find all suitable resources from the both the local and remote data repositories, semantic similarity helps users to find the ‘best’ resource with confidence.

In the end, two research projects, USGS Virtual Arctic Spatial Data Infrastructure and ESIP Semantic Web Testbed, were developed by employing the proposed methodologies. Arctic researchers benefit from the advanced technologies in conducting more efficient and accurate scientific analysis. The ESIP Semantic Web Testbed provides Earth Scientists an integral tool for registering, searching and visualization of Earth Science data and services.

6.2 Future Research

Several problems remain unsolved and need further study. As geospatial data and services increase exponentially on the Web, it will become increasing difficult to locate relevant resources in time-sensitive situations. This requires the search engine to be more flexible and intelligent. This dissertation only solves the problem within a specific domain (hydrology) and the whole search process is not fully automated.

Although the proposed crawler has identified a significant number of live geospatial services, this number still under represents the actual number. This occurs because a fair number of services reside in the deep Web [150], such as those behind firewalls or hosted on private servers [151]. Supporting technologies, such as Universal Description Discovery and Integration (UDDI) and Google Sitemap, need to be further investigated.

For semantic reasoning, the developed tools are still not capable of handling natural language search requests. An effective solution to this problem has eluded computer scientists for decades. Combining logical reasoning with other machine learning technologies, e.g. generic algorithms, is one possible solution. The collaborative ontology development prototype introduced in this dissertation should be publicized more widely so that a greater amount of volunteered data from Internet users could be included, increasing the completeness and accuracy of the KB. This paradigm also fits the recent concept of VGI (Volunteer Geographic Information) in the geospatial community [152].

For semantic similarity measurement, the proposed methodology was verified for the hydrology domain. In the future, training data from other domains, such as geology,

biology, and astronomy need to be collected. In addition, the ubiquity of the proposed methodology needs to be validated and promoted. Neural network training is a time consuming process, especially when the training set is large and the underlining pattern is complex. Therefore, how to parallel the algorithm to improve the efficiency for pre-processing is another issue to be studied.

More broadly, how to fuse the proposed approaches to satisfy the requirements of various applications deserves to be discussed.

References

Reference

[1] GVU, 2009. GVU 9th WWW User Survey. http://www.cc.gatech.edu/gvu/user_surveys/survey-1998-04/. Last access: Dec. 2009.

[2] M. McGiboney, 2009. Nielsen announces June U.S. Search Share Rankings. http://en-us.nielsen.com/etc/content/nielsen_dotcom/en_us/home/news/news_releases/2009/august/video_data_may_2009.mbc.87769.RelatedLinks.46207.MediaPath.pdf. Last access Dec. 2009

[3] BDO Consultants, 1998, Elektronische Bestanden Van Het Bestuur, Report prepared for the Dutch Ministery of the Interior. (Electronic files of government)

[4] P. H. Martin, E. J. LeBoeuf, J. P. Dobbins, E. B. Daniel, and M. D. Abkowitz. 2005. Interfacing GIS with water resource models: A state-of-the-art review. J. American Water Resources Assoc. 41(6): 1471-1487.

[5] W. Li, C. Yang and D. Sun, 2008. Mining the correlation of geophysical parameters' contribution to tropical storms through decision-Tree analysis. Computer and Geosciences, 35(2), 309-316.

[6] C.L. Burton, and M.S.Rosenbaum, 2003, Decision support to assist environmental sedimentology modeling. Environmental Geology, 43, pp. 457–465.

[7] T. Rae-dupree, 2006, Dash Navigation - Real-time GPS driving. Business Journal, http://sanjose.bizjournals.com/sanjose/stories/2006/11/20/focus15.html (accessed 23, January, 2008)

[8] E. Peytchev and C.Claramunt, 2001, Inelegant transportation systems and network algorithms: Experiences in building decision support systems for traffic and transportation GIS. In Proceedings of the 9th ACM international symposium on Advances in geographic information systems GIS '01, Georgia, USA, pp. 154-159.

[9] D. Stevens, S. Dragicevic and K. Rothley, 2007, iCity: A GIS-CA modeling tool for urban planning and decision making. Environmental Modeling & Software, 22, pp. 761-773.

[10] E. Klien, M. Lutz and W. Kuhn, 2006, Computers, Environment and Urban Systems, 30(1), 102-123.

[11] I. Rauschert, P. Agrawal, R. Sharma, S. Fuhrmann, I. Brewer, and A. MacEachren, 2002, User interfaces: Designing a human-centered, multimodal GIS interface to support emergency management. In Proceedings of the 10th ACM international symposium on Advances in geographic information systems GIS '02, Virginia, USA, pp. 119 -124.

[12] M. Paul and S.K.Ghosh, 2006. An approach for service oriented discovery and retrieval of spatial data. In: Proceedings of the 2006 international workshop on service-oriented software engineering, E. Di Nitto, R. J. Hall, J. Han, Y. Han, A. Polini, K. Sandkuhl, A. Zisman (Eds.), 88–94.New York: ACM.

[13] D. Filo, J Yang and K Heyman, 1995. Yahoo! Unplugged: Your Discovery Guide to the Web. IDG Books Worldwide, Foster City, California, USA.

[14] M. Gray, 1993. Measuring the Growth of the Web. http://www.mit.edu/people/mkgray/growth/

[15] M. Koster, 1994. ALIWEB-Archie-like indexing in the Web. Computer Networks and ISDN Systems, 175 – 182.

[16] B. Pinkerton, 1994. Finding What People Want: Experiences with the WebCrawler. In Proceedings of the First World Wide Web Conference, Geneva, Switzerland.

[17] R. Seltzer, D.S. Ray and E.j. Ray, 1996. The AltaVista Resolution: How to Find Anything on the Internet. Osborne/McGraw-Hill Berkeley, CA, USA.

[18] M.L. Mauldin, 1997. Lycos: Design choices in an Internet search service. IEEE Expert, pp.8-11.

[19] S. Brin and L. Page, 1998. The anatomy of a large-scale hypertexual Web Search Engine. In: Proc. of the 7th International WWW Conference (WWW 98) Brisbane, Australia, Comput. Networks ISDN System 30 (April 14–18, 1998), pp. 107–117.

[20] S. Mingay, 2007, Green IT: the new industry shock wave. Gartner RAS Core Research Note G00153703, 2, Article 10.

[21] Wikipedia, 2009b. Pagerank. http://en.wikipedia.org/wiki/Pagerank. Last accessed at July 3, 2010.

[22] Avtec, 2007. Google receives 64 percent of all U.S. searchers in Aug. 2007. Search Engine Statistics. http://avtecmedia.com/internet-marketing/internet-marketing-trends.htm. Last accessed at July 3, 2010

[23] T. Peggs, 2009. The Inner Workings of a Realtime Search Engine. OneRiot Whitepaper, available at http://blog.oneriot.com/content/2009/06/the-inner-workings-of-a-realtime-search-engine/ Last accessed at July 3, 2010

[24] P. Buchheit, 2009. The Man Who Made Gmail Says Real-Time Converstation is What's Next. http://www.readwriteWeb.com/archives/the_man_who_made_gmail_says_real-time_conversation.php. Last access at July 3, 2010

[25] Berners-Lee T., Hendler J., and Lassila O., 2001. The Semantic Web, Scientific American 284, 34–43.

[26] G. Klyne and J. J. Carroll. Resource description framework (RDF): Concepts and abstract syntax. http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/, February 2004.

[27] D. Brickley and R.V. Guha (eds.), 2004. RDF Vocabulary Description Language 1.0: RDF Schema. http://www.w3.org/TR/rdf-schema/, Last accessed at: July 1st, 2010.

[28] M. Dean and G. Schreiber, 2004. OWL Web ontology language reference. http://www.w3.org/TR/2004/REC-owl-ref-20040210/, February 2004.

[29] W. Li, C. Yang and B. Zhou, 2008. Internet-based spatial information retrieval. Encyclopedia of GIS 2008: 596-599.

[30] Lskold, 2008. Semantic Search: The Myth and Reality. ReadWriteWeb. http://www.readwriteWeb.com/archives/semantic_search_the_myth_and_reality.php

[31] Wikipedia, 2009a. Semantic Search. http://en.wikipedia.org/wiki/Semantic_search. Last accessed at July 3, 2010.

[32] J. Townley, 2000. The Streaming Search Engine That Reads Your Mind, Streaming Media World, http://smw.internet.com/gen/reviews/searchassociation/index.html. Last accessed at July 3, 2010

[33] R. Guha, R. McCool and E. Miller, 2003. Semantic Search. Proceedings of the WWW2003, Budapest, 2003.

[34] Rocha, D. Schwabe and M.P. de Aragao, 2005. A hybrid Approach for Searching in the Semantic Web.

[35] E. Hatcher and O. Gospodnetic, 2004. Lucene in Action. Action series. Manning Publications Co. Greenwich, CT, USA.

[36] R. Raskin and M. Pan, 2005. Knowledge representation in the semantic Web for Earth and environmental terminology (SWEET), Computer & Geosciences, 31(9), 1119-1125.

[37] R. Ramachandran, S. Movva, P. Cherukuri and S. Graves, 2006. Noesis—An Ontology-Based Semantic Search Tool and Resource Aggregator. Geoinformatics 2006, pp. 35.

[38] F. Baader, D. Calvanese and D.L. McGuinness, 2003. The description logic handbook: theory, implementation, and applications.

[39] P. Fox, D. L. McGuinness, L. Cinquini, P. West, J. Garcia, J.Benedict and D. Middleton, 2008. Ontology-supported Scientific Data Frameworks: The Virtual Solar-Terrestrial Observatory Experience, Computers and Geosciences, Special issue on Geoscience Knowledge Representation for Cyberinfrastructure.

[40] Ben-Yitzhak, N. Golbandi, N. Har'El, R. Lempel, A. Neumann, S. Ofek-Koifman, D. Sheinwald, E.Sheinwald, E. Shekita, B. Sznajder and S. Yogev, 2008. In: Proceedings of the international conference on Web search and Web data mining, Palo Alto, California, USA, 33-44.

[41] W. Dakka, R. Dayal, and P. Ipeirotis, 2006. Automatic discovery of useful facet terms, in ACM SIGIR 2006 Workshop on Faceted Search.

[42] M. Kehoe, 2005. Parametric Search, Faceted Search, and Taxonomies, NIE Enterprise Search, 2(6). http://www.ideaeng.com/tabId/98/itemId/86/Parametric-Search-Faceted-Search-and-Taxonomies.aspx Last accessed at July 3, 2010

[43] P. Marshall, S. Herman and S. Rajan, 2006. In search of more meaningful search, Serials Review, 32(3).
[44] Y. Wang and P. Jhuo, 2009. A Semantic Faceted Search with Rule-based Inference. In: Proceedings of the International MultiConference of Engineers and Computer Scientists 2009 Vol I IMECS 2009, March 18 - 20, 2009, Hong Kong.
[45] G. Salton and M.Mcgill, 1983. Introduction to Modern Information Retrieval, New York: McGraw-Hill.
[46] C. Buckley, J. Allan, G. Salton and A. Singhal. Automatic query expansion using SMART : TREC 3. In D. K. H#rman, editor, Proceedings of the Thzrd Text REtrieval Conference (TREC-3), pages 69- 80. NIST Special Publication 500-225, April 1995.
[47] C. Zhai, X. Tong, N. Milic-Frayling and D. A. Evans, 1997. Evaluation of syntactic phrase indexing - CLARIT NLP track report. In: D. K. Harman (ed.), The Fifrh Texr Retrieval Conference (TREC-5). NIST Special Publication.
[48] S. Deerwesters, S. Dumais, G. Furnas, T. Landauer and R. Harshman, 1990. Indexing by latent semantic analysis, J. American Society for Information Science, 41(1990), 391-407.
[49] M.W. Berry, S.T. Dumais and G.W. O'Brien, 1995. Using linear algebra for intelligent information retrieval, SIAM Rev. 37(1995), 573-595.
[50] M. W. Berry, Z. Drmac and E. R. Jessup, 1999. Matrices, Vector Spaces, and Information Retrieval. SIAM Review, 41(2), 335-362.
[51] T.K. Landauer, P.W. Foltz and D. Laham, 1998. Introduction to Latent Semantic Analysis. Discourse Processes, 25, 259-284.
[52] NASA, 2007, Earth System Science Data Resources: tapping into a wealth of data, information, and services. National Aeronautics and Space Administration, 65p.
[53] D. Nebert , ed., 2004. Developing spatial data infrastructures: the SDI cookbook, Version 2.0, Global Spatial Data Infrastructure, 171pp. Available online at: http://www.gsdi.org/docs2004/Cookbook/cookbookV2.0.pdf (accessed 15th January 2010).
[54] de La Beaujardiere, J. (Ed.), 2004, Web Map Service Implementation Specifications, Version 1.3. OGC Document Number: 04-024, Open Geospatial Consortium, U.S., 85pp.
[55] Vretanos, P.A. (ed.), 2005, Web Feature Service Implementation Specifications, Version 1.1.0. OGC Document Number: 04-094, Open Geospatial Consortium, 131pp.
[56] Whiteside, A. and Evans, J. (Eds), 2006, Web Coverage Service Implementation Specifications, Version 1.1.0. OGC Document Number: 06-083r8, Open Geospatial Consortium, U.S., 143pp.
[57] Yang, C. and Tao, C. V., 2006, Distributed Geospatial Information Service (Distributed GIService). In Frontiers of Geographic Information Technology, S. Rana and J. Sharma (Eds), pp. 103-120 (New York, US: Springer).

[58] Yang, C., Li, W., Xie, J. and Zhou, B., 2008, Distributed geospatial information processing - sharing distributed geospatial resources to support the Digital Earth. International Journal of Digital Earth, 1, pp. 259-278.

[59] Alameh, N., 2003, Chaining geographic information Web services. Internet Computing, 7, pp. 22-29.

[60] Rajasekaran P., Miller J. A., Verma K., and Sheth A. P., 2004, Enhancing Web services description and discovery to facilitate composition. In Proceedings of First International Semantic Web Services and Web Process Composition Workshop, San Diego, California, USA.

[61] D. Roman, U. Keller, H. Lausen, J. D. Bruijn, R. Lara, M. Stollberg, A. Polleres, C. Feier, C. Bussler and D. Fensel1, 2005, Web service modeling ontology. Applied Ontology, 1, pp. 77-106.

[62] C. Yang, W. Li, J. Xie and B. Zhou, 2008. Distributed geospatial information processing - sharing distributed geospatial resources to support the Digital Earth, International Journal of Digital Earth, 1(3), 259-278..

[63] A. Dogac, Y. Kabak and G.B. Laleci, 2006, ebXML Registry Profile for Web Ontology Language (OWL). Organization for the Advancement of Structured Information Standards (OASIS), 75pp.

[64] K. Stock (ed.), 2009, OGC Catalogue Services – OWL Application Profile of CSW. Document Number: 09-010, Open Geospatial Consortim Inc., 70pp.

[65] U. Keller, R. Lara and A. Polleres (Eds.), 2004, WSMO Web service discovery. In WSMO Web Service Discovery Working Draft D5.1v0.1, U. Keller, R. Lara, A. Polleres, I. Toma, M. Kifer and D. Fensel (Eds), pp. 13-15.

[66] M. Egenhofer, 2002, Toward the semantic geospatial Web. In Proceedings of the 10th ACM International Symposium on Advances in Geographic Information Systems, Virginia, USA, pp. 1-4.

[67] J.T. Sample, R. Ladner, L. Shulman, E. Ioup, F. Petry, E. Warner, K. Shaw and F.P. McCreedy, 2006, Enhancing the US Navy's GIDB Portal with Web Services. Internet Computing 10, pp. 53-60.

[68] G. Singh, S. Bharathi, A. Chervenak, E. Deelman, C. Kesselman, M. Manohar, S. Patil and L. Pearlman, 2003, A metadata catalog service for data intensive applications. In Proceedings of the 2003 ACM/IEEE conference on Supercomputing, Washington, DC, USA, pp. 33.

[69] X. Ma, G. Li, K. Xie and M. Shuai, 2006, A Peer-to-Peer approach to geospatial Web service discovery. In Proceedings of 1st international conference on Scalable information system, Hong Kong, China, Article No. 53.

[70] E. Al-Masri and Q.H. Mahmoud, 2007, Interoperability among service registry standards. Internet Computing. 11, pp. 74-77.

[71] S. Ran, 2004, A model for Web services discovery with QoS. SIGEcom Exchanges. 4, pp. 1-10.

[72] M. Reichardt, 2005, GSDI depends on widespread adoption of OGC standards. In Proceedings of the FIG Working Week and GSDI8: From Pharaohs to Geoinformatics, Cairo, Egypt.

[73] K. Wöber, 2006, Domain specific search engines. In Destination Recommendation Systems: Behavioral Foundations and Applications, D. R. Fesenmaier, K. Wöber, and H. Werthner (Eds). (Wallingford, UK: CABI)

[74] D.R. Fesenmaier, K.W. Wöber and H. Werthner, 2006, Destination Recommendation Systems: Behavioral Foundations and Applications, pp. 205-220 (Oxford, UK: Oxford University Press).

[75] R. Steele, 2001, Techniques for specialized search engines. In Proceedings of the International Conference on Internet Computing, IC'2001, Las Vegas, Nevada, USA, pp. 25-28.

[76] A. Schutzberg, 2006, Skylab Mobilesystems crawls the Web for Web Map Services. OGC User, 8, pp. 1-3.

[77] W. Li, C. Yang and C. Yang, 2010. An active crawler for discovering geospatial Web services and their distribution pattern – A case study of OGC Web Map Service. International Journal of Geographic Information Science, 24(8), 1127-1147.

[78] W. Li, 2007, Interoperability based spatial information retrieval and Ontology based semantic search. Thesis (M.Sc.), Institute of Remote Sensing Applications, Chinese Academy of Sciences, Beijing, China, 84 pp.

[79] S. Dill, R. Kumar, K. S. Mccurley, S. Rajagopalan, D. Sivakumar, and A. Tomkins, 2002, Self-similarity in the Web. ACM Transactions on Internet Technology, 2, pp. 205–223.

[80] B. Sergey and L. Page, 1999. The anatomy of a large-scale hypertextualWeb search engine. Computer Networks and ISDN Systems, 30, 107–117.

[81] M. D. Kunder, 2009, The size of World Wide Web. http://www.worldwideWebsize.com/, last access: July, 2010.

[82] S. Mingay, 2007, Green IT: the new industry shock wave. Gartner RAS Core Research Note G00153703, 2, Article 10.

[83] K. Christopher, 1995. 'Reasoning'. In Ted Honderich (ed.), The Oxford Companion to Philosophy. Oxford: Oxford University Press: p. 748

[84] P. Lipton, 2001. Inference to the Best Explanation, London: Routledge. ISBN 0-415-24202-9.

[85] B. C. Hoekenga, 2007. Mind over machine : what Deep Blue taught us about chess, artificial intelligence, and the human spirit. Master thesis, 49pp. MIT DSpace: http://hdl.handle.net/1721.1/42144 (Last accessed Jan. 11 2010)

[86] H. Gilbert,1965. The Inference to the Best Explanation, The Philosophical Review 74:1, 88-95.

[87] P. Johnson-Laird, and R. M. J. Byrne, 1991. Deduction, Psychology Press, ISBN 9780863771491

[88] J. R. Josephson and S. G. Josephson (eds.), 1995. Abductive Inference: Computation, Philosophy, Technology, Cambridge University Press, Cambridge, UK.

[89] V. F. Hendricks, 2005. Thought 2 Talk: A Crash Course in Reflection and Expression, New York: Automatic Press / VIP, SBN 87-991013-7-8.

[90] E. Amir and S. Mcllraith, 2005. Partition-based logical reasoning for first-order and propositional theories. Artificial Intelligence Conference on Temporal Logic, pp. 17- 25.

[91] V. Haarslev and R. Moeller. 2003. Racer: A core inference engine for the Semantic Web. In 2nd International Workshop on Evaluation of Ontology-based Tools (EON-2003), Sanibel Island, FL.

[92] J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, K. Wilkinson, 2004. Jena – Implementing the Semantic Web Recommendations. In: Proceedings of the Thirteenth International World Wide Web Conference (WWW2004), 2004.

[93] D. Tsarkov and I. Horrocks. 2006. FaCT++ description logic reasoner: system description, 2006. In: Proceedings of the International Joint Conference on Automated Reasoning (IJCAR 2006). Lecture Notes in Artificial Intelligence, vol. 4130, pp. 292–297. Springer, Berlin Heidelberg New York.

[94] E. Sirin, B. Parsia, B. G. Cuenca, A. Kalyanpur, and Y. Katz 2007. Pellet: a practical OWL-DL reasoner. J. Web Semantics 5(2), 51–53.

[95] J. Holland, 1975. Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor.

[96] D. E. Goldberg, 1989. Genetic algorithms in search, optimization, and machine learning. Addison-Wesley.

[97] G. Chierchia, 1999. Linguistics and Language. In Wilson, R.A. and Keil, F.C. (Eds.): The MIT Encyclopedia of the Cognitive Sciences, pp. xv-xxxvii, Massachusetts: The MIT Press.

[98] K. Thomson, 2009. Dwelling on ontology – semantic reasoning over topographic maps. Doctoral thesis, University College London.

[99] F., Sowa, 2000. Knowledge Representation: Logical, Philosophical, and Computational Foundations. Pacific Grove, CA: Brooks Cole Publishing.

[100] V. Rus, 2002. Logic Form for WordNet Glosses. Ph.D. thesis, Southern Methodist University.

[101] IEP, 2010. The Internet Encyclopedia or Philosophy. http://www.iep.utm.edu/prop-log/. Last access at July 1, 2010.

[102] T. Gruber, 1993. A translation approach to portable ontology specification. ACMD Knowledge Acquisition, Special issue: Current issues in knowledge modeling, 6(2), 199-220.

[103] M.A. Latre, J. Lacasta, E. Mojica, J. Nogueras-Iso and F.J. Zarazaga-Soria, 2009. An approach to facilitate the integration of hydrological data by means of ontologies and multilingual thesauri. In: Proceedings of 12th AGILE International Conference on Geographic Information Science – Advanced in GIScience, Hannover, Germany, pp.155-171.

[104] D. Tarboton, J. Horsburgh, D. Maidment and B. Jennings, 2006. CUAHSI community observations data model working design specifications document –V3.1. CUAHSI org. http://giscenter.isu.edu/training/pdf/geotech_semi [accessed 05 May, 2010]

[105] L. Olsen, 2001. Supporting interagency collaboration through the Global Change Data and Information System (GCDIS). In: Proceedings of the American Meteorological Society, Long Beach, CA, pp. 348–351.
[106] GEMET, 1999. General European Multi-lingual Environmental Thesaurus, European Environmental Agency. http://www.eea.eu.int, [accessed 05 May, 2010]
[107] F. Steimann and W. Nejdl, 1999. Modellierung und Ontologie. Technischer Bericht, Universität Hannover, Institut für Rechnergestützte Wissensverarbeitung, http://www.kbs.unihannover.de/Arbeiten/Publikationen/1999/M&O.pdf
[108] W. Li, C.Yang and R. Raskin, 2008, A semantic enhanced search for spatial Web portals. In Technical report of Semantic Scientific Knowledge Integration of AAAI Spring Symposium, pp. 47- 50, (Merlo Park, CA: AAAI Press).
[109] W. Li and C. Yang, 2008. A Semantic Search Engine for Spatial Web Portal. In: Proceedings of IEEE International Geosciences and Remote Sensing Symposium, IGARSS08, Boston, US, 2008.
[110] D. Klein and C. D.Manning, 2003. Fast exact inference with a factored model for natural language parsing. In: Proceedings of Advances in Neural Information Processing Systems 15, Cambridge, MA, MIT Press, pp. 3-10.
[111] E. Prud'hommeaux and A.Seaborne, 2005. SPARQL query language for RDF. World Wide Consortium. http://www.w3.org/TR/2005/WD-rdf-sparql-query-20050217/., [accessed 05 May, 2010]
[112] C. Kebler, 2007. Similarity Measurement in Context. Context 2007, LNAI 4635, 277-290.
[113] T. Schwering and W. Kuhn, 2009. A Hybrid Semantic Similarity Measure for Spatial Information Retrieval. Spatial Cognition and Computation. 9(1), 30-63.
[114] A. Sheth, 1999. Changing Focus on Interoperability in Information Systems: From System Syntax, Structure to Semantics. Interoperating Geographic Information Systems (eds), 5-30.
[115] M.A. Rodriguez and M.J. Egenhofer, 2004. Comparing geospatial entity classes: an asymmetric and context-dependent similarity measure. IJGIS, 18(3), 229-256.
[116] P. Santos, N. Bennett and G. Sakellariou, 2005. Supervaluation Semantics for an Inland Water Feature Ontology. Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence,Edinburgh, Scotland, July 30 - August 5, 2005.
[117] P. Resnik, 1999. Semantic Similarity in Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. Journal of Artificial Intelligence Research 11, 95-130.
[118] M. A. Eshera and K. S. Fu, A graph distance measure for image analysis. IEEE Trans. Systems, Man, Cybern. 14 (1984), pp. 398–408.
[119] R. Rada, H. Mili, E. Bicknell, and M. Blettner,1989. Development and application of a metric on semantic nets. IEEE Transaction on Systems, Man, and Cybernetics, 19(1):17–30.

[120] D. Lin, 1998. An Information-Theoretic Definition of Similarity. Proceedings of the 15th International Conference on Machine Learning, 296-304.
[121] A. Tversky, 1977. Features of Similarity. Psychological Review, 84(4), 327-352.
[122] H. Rubenstein and J. B. Goodenough, 1965. Contextual Correlates of Synonymy. Communications of the ACM 8(10): 627-633.
[123] J. Amini, 2008. Optimum Learning Rate in Back-Propogation Neural Network for Classification of Satellite Images (IRS-1D). Scientia Iranica, 15(6), pp.558-567.
[124] Y. Tan, 2002. A neural network approach for signal detection in digital communications. The Journal of VLSI Signal Processing, 32(1), 45-54.
[125] R. Rada, H. Mili, E. Bicknell and M. Blettner, 1989. Development and application of a Metric on Semantic Nets. IEEE Transactions on System, Man and Cybernetics, 19(1), 17-30.
[126] D. White, L. Hinzman, L. Alessa, J. Cassano, M. Chambers, K. Falkner, J. Francis, W.J. Gutowski Jr., M. Holland, R.M. Holmes, H. Huntington, D. Kane, A. Kliskey, C. Lee, J. McClelland, B. Peterson, T.S. Rupp, F. Straneo, M. Steele, R. Woodgate, D. Yang, K. Yoshikawa and T. Zhang, 2007. The arctic freshwater system: changes and impacts. Journal of Geophysical Research, 112 (2007), p. G04S54 10.1029/2006JG000353.
[127] P. Wadhams and W. Munk 2004. Ocean freshening, sea level rising, sea ice melting. Geophys. Res. Lett. 31, (L11311), doi:101029/2004GLO20039.
[128] A. Mehta, 2008. North Pole may be ice-free for first time this summer. National Geographic News. http://news.nationalgeographic.com/news/2008/06/080620-north-pole.html. [accessed 05 May,2010]
[129] D. Nebert, 2004, Developing Spatial Data Infrastructures: The SDI Cookbook, Version 2.0, Global Spatial Data Infrastructure, D. Nebert (Eds), 171pp.
[130] A. Rajabifard, M. F. Feeney and I. P. Williamson, 2003. Future directions for SDI development. International Journal of Applied Earth Observation and GeoInformation, 4(1), 11-12.
[131] P. Holland, M.E. Reichardt, D. Nebert, S. Blake, D. Robertson, 1999. The global spatial data infrastructure initiative and its relationship to the vision of a digital earth. In: Proceedings of the International Symposium on Digital Earth, Beijing, China, pp.1-7.
[132] L. Bernard, I. Kanellopoulos, A. Annoni and P. Smits, 2004. The European geoportal—one step towards the establishment of a European Spatial Data Infrastructure. Computers, Environment and Urban Systems, 29(1), 15-31.
[133] D.J. Coleman and D.D. Nebert, 1998. Building a North American spatial data infrastructure. Cartography and Geographic Information Science, 25(3), 151-160.
[134] M. Craglia and A. Annoni, 2006. INSPIRE: An innovative approach to the development of Spatial Data Infrastructure in Europe. In: Onsrud H. (Ed.) Research and Theory in Advancing Spatial Data Infrastructure, ESRI press Redlands, pp. 93-105.

[135] Y. Georgiadou, S.K. Puri and S. Sahay, 2005. Towards a potential research agenda to guide the implementation of Spatial Data Infrastructures—A case study from India. International Journal of Geographic Information Science, 19(10), 1113-1130.
[136] S. Jacoby, J. Smith, L. Ting and I. Williamson, 2002. Developing a common spatial data infrastructure between state and local government--an Australian case study. International Journal of Geographic Information Science, 16(4), 305-322.
[137] A. Rajabifard, T. O. Chan and I. P. Williamson, 1999. The nature of regional spatial data infrastructures. In: Proceedings, of the 27th Annual Conference of AURISA, Blue Mountains, New South Wales.
[138] W. Li, C. Yang, D. Nebert, R. Raskin, P. Houser, H. Wu and Z. Li. Semantic-based Web service discovery and chaining for building an Arctic Spatial Data Infrastructure. Computers & Geosciences (to appear).
[139] A. Sheth, C. Ramakrishnan and C. Thomas, 2005. Semantics for the Semantic Web: The Implicit, the Formal and the Powerful. Int'l Journal on Semantic Web & Information Systems, 1(1), 1-18.
[140] S. Staab and A. Maedche. Knowledge Portals --- Ontologies at work. AI Magazine, 21(2), PP.63-75.
[141] B. Omelayenko, 2001. Learning of Ontologies for the Web: the Analysis of Existent approaches. In Proceedings of the International Workshop on Web Dynamics, 2001.
[142] B. Hammond, A. Sheth, and K. Kochut, 2001. Semantic Enhancement Engine: A Modular Document Enhancement Platform for Semantic Applications over Heterogeneous Content, in Real World Semantic Web Applications, V. Kashyap and L. Shklar, Eds., IOS Press, pp. 29-49.
[143] S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran and T. Kanungo, S. Rajagopalan, A. Tomkins, J.A. Tomlin, and J. Y. Zien, 2003. Semtag and seeker: bootstrapping the semantic Web via automated semantic annotation, In Proceedings of the Twelfth International Conference on World Wide Web, 178-186.
[144] A. Patil, S. Oundhakar, A. Sheth and K.Verma, 2004. METEOR-S Web service Annotation Framework, Proceeding of the World Wide Web Conference, New York, NY, pp. 553-562.
[145] A. Rector and I. Horrocks, 1997. Experience building a large, re-usable medical ontology using description logic with transitivity and concept inclusion. Proceedings of the Workshop on Ontological Engineering, AAAI, 1997.
[146] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin and G. Sherlock, 2000. Gene Ontology: tool for the unification of biology. Nature Genetics 25, 25-29.

[147] J.E. Gordon, 1988. Chemical inference, 3. formalization of the language of relational chemistry: ontology and algebra, Journal of Chemical Information & Computer Sciences, 28(2), 100-115.
[148] W. Li, C. Yang and R. Raskin, 2010. Semantic Integration of Geoscientific Data through ESIP Semantic Web Testbed, In: Proceedings of IEEE International Conference of Remote Sensing and Geosciences, Hawaii, USA.
[149] W. Li and C. Yang, 2009. A semantic meta-catalogue for intelligent geographic information discovery. In: Proceedings of Geoinformatics 2009, Fairfax, VA.
[150] S. Lawrence and C. L. Giles, 1999, Accessibility of information on the Web. Nature, 400, pp. 107 - 109.
[151] P. Ramsey, 2005, A Census of Public OGC Web Services. Presentation to the OGC Planning Committee, Refractions Research.
[152] M.F. Goodchild, 2007. Citizens as sensors: the world of volunteered geography. GeoJournal 69(4): 211-221.

Curriculum Vitae

Wenwen Li has been a PhD student in the Center for Intelligent Spatial Computing at George Mason University (GMU) since 2006. Her major is Earth Systems and Geoinformation Sciences. Before joining GMU, she received a B.S. degree in Computer Science and Technology from Beijing Normal University, China, in 2004 and a M.S. degree in Signal and Information Processing from Chinese Academy of Sciences, China in 2007. Her research focuses on semantic search, distributed geoinformation processing and geospatial data mining.
`