A CONCEPT SPACE APPROACH TO SEMANTIC EXCHANGE

by

Tobun Dorbin Ng

A Dissertation Submitted to the

COMMITTEE on BUSINESS ADMINISTRATION

In Partial Fulfilment of the Requirements
For the Degree of

DOCTOR OF PHILOSOPHY
WITH A MAJOR IN MANAGEMENT

In the Graduate College

THE UNIVERSITY OF ARIZONA

2 0 0 0

STATEMENT BY AUTHOR

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

Brief quotations from this dissertation are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the copyright holder.

SIGNED: ______________________________

ACKNOWLEDGMENTS

It has been an exceptional journey - one that I had never dreamed of. My research career was started not long after I became an MIS undergraduate student in 1989, when I attended Dr. Hsinchun Chen's first class at The University of Arizona. I am deeply indebted to Dr. Chen for his encouragement, advice, mentoring, and research support throughout my undergraduate, master's, and doctoral studies. I also truly appreciate his patience and tolerance during my numerous mishaps. This dissertation is part of the research carried out through his vision throughout last ten years.

I am fortunate to have the opportunity to work with a group of energetic people in Dr. Chen's AI Lab. I have enjoyed every moment that we have worked together including all those late night lab activities. All former and present AI Lab members have taught me many things about life. I appreciate all their friendships and their collective encouragement to finish this dissertation. I want especially to thank Chienting Lin, Thian-Huat Ong, Marshall Ramsey, Yohanes Santoso, Harry Li, Samuel Yim, and Chris Schuffels for sharing their technical wisdom, and to thank Andrea Houston, Joanne Martinez, Kris Tolle, and Rosie Hauck for sharing their research ideas.

Finally, it is impossible to have my research career without my parents' love and support, as well as my family and friends' encouragement. This dissertation is dedicated to them.

To all of you, thank you.

DEDICATION

This dissertation is a result of collective efforts from my parents, family, and friends. They often give me valuable advice, reminders, and wisdom about life, when I had a hard time understanding. However, they always give me room to explore and make mistakes.

This dissertation is dedicated to my parents, who have given me all their love and support and let me freely do whatever I want. Without them, there is no way I could possibly have accomplished this. Their understanding on the value of education is truly beyond my comprehension. I am just a lucky beneficiary. Along the way, my sister Karen and brother Gavin have shared their caring thoughts.

I have also been very lucky to have my uncle and aunt, Bob and Hannah Ng, and my cousins, Belinda and Gigi, throughout my stay in Arizona. They have helped me learn about life and live a good one. They have also given me comfort and advice whenever I needed them.

Throughout my ten plus years in Arizona and Dr. Hsinchun Chen's AI Lab, I have seen many people come and go; however, I am very grateful that many of them have taken me as their friend. They have taught me how to live, love, and feel; these, in turn, become the catalyst to my desire to finish this dissertation.

I truly thank all of them from the bottom of my heart.

We are loved beyond our capacity to comprehend. - Jewel Kilcher

TABLE OF CONTENTS

TABLE OF CONTENTS – *Continued*

TABLE OF CONTENTS – *Continued*

TABLE OF CONTENTS – *Continued*

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

This dissertation work investigates the use of information technologies that clarify semantic meaning to help users elaborate their information needs by providing library-specific knowledge to the information seeking process. The research involved two interdependent semantic technologies: concept space consultation and library-specific, domain-specific, automatically generated concept spaces.

The concept space consultation phase used spreading activation algorithms - branch-and-bound and Hopfield net algorithms - to explore knowledge sources in specific domains. This research demonstrated the comparable effectiveness of exploration of a library database using a man-made classification scheme and thesaurus as opposed to an automatically generated concept space. The results showed that the use of spreading activation algorithms identified more relevant concepts than the use of the manual browsing method.

The concept space technique automatically identifies and extracts concept from a library collection while at the same time computing the strength of associations between concepts. This research demonstrated that the concept space technique was able to create human-recognizable concepts and their associations. In addition, the technique could be scaled to generate very large library-specific concept spaces for a very large underlying library collection.

Moreover, the interdependent use of both semantic technologies creates a semantic medium for users and library-specific knowledge sources to exchange content with context - context in user information need and that in corporeal knowledge.

CHAPTER 1

INTRODUCTION

This dissertation work investigated the use of information technologies that clarify semantic meaning to help users elaborate their information needs by specifying their library-specific knowledge during the information seeking process. That process starts with expression of a user information need, which makes it dynamic, cognitive, and user-oriented. However, existing information retrieval processes take only the end result of the cognitive aspect as query input and implicitly assume that the query input is the true (or almost true) representation of the user information need. Is this assumption always true?

The research in knowledge discovery, data mining, and machine learning has demonstrated the ability to generate knowledge of underlying data collections. In addition to automatic computational methods, extensive human effort has been invested in building domain-specific thesauri and classification schemes. However, all these semantic-bearing entities are targeted to represent knowledge coverage in particular domains and to capture semantic relationships between identified concepts. Can these knowledge sources adequately serve users' information needs?

When the information seeking process reaches the stage of retrieving information from search engines, users must perform two tasks - making a query and evaluating a list of retrieved documents. If users are not satisfied with the results, they will repeat the same two tasks with different queries or they will give up their searches. This "blackbox" information retrieval approach has existed since the beginning of the information retrieval field in the 1960s and has since been popularized by the Internet. Nowadays, users are very familiar with what to expect from search engines - a possibly long ranked list of documents that they may never go through. In addition, in information retrieval practices, a common belief has been created that "relevant" documents appear only at the top 10 or 20 of ranked retrieved result.

The role of users in this "blackbox" approach focuses on evaluating retrieved documents for their relevancy. This approach downplays, if it does not ignore, the necessity of an information need having been fully expressed. This dissertation work adopted a user-centric and interactive approach to helping users elaborate their information needs with library-specific knowledge and simultaneously gain insight into a library's offerings related to their information needs. Through semantic communication between users, users are enabled to express their information needs in the context of the library-specific knowledge of the target information source. Shared contextual information also gives users knowledge of the potential value of retrieved information. Under this approach, the information seeking process becomes a semantic journey from the initial expression of a user's information need to acquisition of knowledge of relevant information rather than a purely computational operation.

In order to explore semantic exchange between users and collections, this dissertation research focused on two areas: interactive consultation with knowledge sources and automatic generation of semantic-bearing knowledge sources from corresponding libraries. Concept space consultation demonstrates a semantic exchange between users and knowledge sources during query expression. Concept space generation shows how large-scale semantic-bearing knowledge sources can be automatically generated.

CHAPTER 2

LITERATURE REVIEW

This chapter reviews previous research on two aspects of information searching: library collection and user information need. The first section examines various concerns and techniques employed to characterize an individual document object or a collection of such document objects. The core of an information retrieval system is its analytical capability to dissect, digest, and derive knowledge from underlying records. The second section is devoted to investigating user information needs with respect to the information seeking process. In general, information retrieval systems are able to manage underlying information and even derived knowledge. However, such systems' capability remains static in comparison with the dynamic nature of user needs.

2.1 Static Nature of Knowledge in Library Collection

Recent development in research and technology has advanced information retrieval systems to enable them to handle multimedia objects such as images (Ma and Manjunath, 1998), audio (Witten et al., 1999) and video (Wactlar et al., 1999). However, the discussion in this paper mainly focuses on textual documents, which include bibliographical records and full-text articles, although it demonstrates some parallelism with analysis of multimedia systems.

Two levels of document analysis are commonly studied and performed by information scientists and practitioners. The next sub-section reviews how a single document is characterized, while the second sub-section depicts how a document collection is analyzed. Instead of focusing on documents, the last sub-section examines concepts or terms existing in documents.

2.1.1 Characterizing Document Objects

2.1.1.1 Theory of Indexing

The dual purposes of indexing an document are to represent a lengthy and structureless textual record by a set of indexes (*atomic elements*) and to access a set of textual records through their indexes (Salton, 1975). Research on indexing has been focused on defining a set of *good* index terms as well as assigning a set of *good* index terms to a particular document. Ideally, the choice of such *good* index terms should collect all relevant documents to yield high recall and simultaneously distinguish them from irrelevant ones in order to give high precision. However, in reality, the choice of index terms always exhibits the phenomenon of the well-known inverse relationship between recall and precision (Salton, 1975). Achieving high recall is generally at the cost of low precision and vice versa.

Lancaster has shown that the rate of growth of information continues at an exponential pace, while the corresponding rate of growth over the same period of time for number of concepts (index terms) converges logarithmically (Lancaster, 1986). Chen has summarized this phenomenon as a logarithmic vocabulary growth principle that sheds light on the *information overload* problem (Chen, 1994). Nonetheless, as the growth of concepts moves farther into the flat region of the plateau, the once manageable volume of concepts led by the logarithmic vocabulary growth makes it difficult to discriminate information related to each concept, which increases at the same exponential pace. This phenomenon can be easily demonstrated by using a web search engine, which returns hundreds of thousand or more web pages from a simple query (Kirsch, 1998).

Based on the document frequency of index terms, Salton has suggested a model for construction of *good* index terms (Salton, 1975), (Salton and Yu, 1973). The model divides terms into three groups according to their document frequencies: low, medium, and high. Given a document collection, all index terms can be ranked by their document frequencies and listed from the left (low frequency) to the right (high frequency). *Good* terms fall into the medium document frequency range. Terms in the high frequency end are considered the *worst* index terms because they do not have discriminating power. Terms in the low frequency end are called *poor* index terms characterized by poor performance on recall. Even though it was developed with two small collections of 450 and 1,400 documents each, the model created by Salton (1975) generally holds true with the World Wide Web's so far approximately 100 million web pages (Kirsch, 1998), (Schwartz, 1998). Of course, the document frequency range for *good* index terms varies between different document collections. However, not much research has been done to investigate what a *good* range would be. Instead, many search engines, especially web search engines, have adotped various strategies to rank hundreds of thousand retrieved documents and give the best 10 or 20 to users (Schwartz, 1998).

With his model for *good* index terms, Salton has proposed two different mechanisms to convert the *worst* or *poor* terms into *good* ones (Salton, 1975). The purpose is to make a collection consistently made up of *good* index terms. The first mechanism is called the *right-to-left phrase construction.* The idea is to transform high frequency terms (on the right end of the document frequency range) into *units* with lower frequency (toward the left to the middle) in order to improve their precision. Such desirable units are *term phrases.* A classical method to do so is to generate *phrases* consisting of several combined terms (Salton, 1988). A restricted, practical, and automatic version of this classical method is to use consecutive adjacent words to form *phrases* (Chen and Lynch, 1992). For example, in a computer science collection, the terms *program* and *language* may be insufficiently specific, particularly when assigned to a large proportion of the documents in a collection. The phrase *programming language* is more specific and may, when assigned to the documents, lead to improved precision output.

The second mechanism is called the *left-to-right thesaurus transformation* (Salton, 1975). The goal is to transform low frequency terms (on the left side) into *units* of higher frequency (toward the right to the middle) in order to improve their recall. Such units are generated by grouping a number of the low-frequency entities into classes. The term classes are then characterized by frequency properties equivalent to the sum of the frequencies of the individual components. A classical way of combining individual terms into classes is by means of a *thesaurus*. Such a thesaurus specifies a grouping of the vocabulary in which items included in the same class are normally considered to be related in some sense – for example, by being synonymous, or by exhibiting closely similar content characteristics. The success of this method relies heavily on the availability of a *good* thesaurus in a given document collection or domain for a given time frame. In practice, low frequency index terms in a very large collection such as World Wide Web, indeed, give quite desirable performance in terms of precision, with the trade off of forgiving and forgettable document recall. Such low frequency is considered to be in hundreds range, compared with the usual hundreds of thousands range.

In addition to size of collection, number of index term assignment has a direct impact on constructing an index having a *good* range. Traditionally, content providers hire human indexers to assign three to six index terms to a document after reading it, hoping of calibrate the choice/usage of controlled vocabularies and the accountability of each index term. Over time, as information technology becomes more capable and affordable, the number of index terms assigned to each document increases. A certain number of free-text term phrases are included as index terms. An extension of free-text indexing, full-text indexing, is available in some information retrieval systems.

2.1.1.2 Manual Indexing

Manual indexing is part of the summarization process professional indexers typically perform on journal articles (Endres-Niggemeyer and Neugebauer, 1998). The whole process requires a expert summarizer to read an article and perform two tasks: abstracting and indexing. Abstracting includes reading, taking notes, drafting an abstract, revising the draft, and writing the final version (Rowley, 1988). The task of indexing is tightly associated with the classification process, which also involves subject analysis, translation into the indexing language, and construction of a register entry (Langridge, 1989). While abstacting produces a relatively long textual summary in natural language format with sentences or even paragraphs, indexing gives a list of discrete term phrases or concepts to represent core ideas in an article and classification assigns one or more artificial codes according to a classficiation scheme. Each summarized record becomes an entry to a bibliographic information system.

Manual indexing is very cognitively intensive but mechanical (Endres-Niggemeyer and Neugebauer, 1998). A study by Endres-Niggemeyer and Neugebauer found that a well-trained indexer deals with each article independently, reading through it only once, an indexer writes down what is noteworthy and creates an abstract. However, the study falls short on describing in detail how index terms are selected by an expert indexer.

In the field of information science, Bates summarizes a consistent and historical phenomenon that indexers simply index what is in the record (Bates, 1998), directly reflecting the fact that a document is known and visible to an indexer. Factual information can be checked directly and immediately to create an absolutely accurate bibliographic record. In addition, indexers are trained to use a specific indexing system and vocabulary, generally establishing rules for resolving debatable situations such as which term is to be used rather than the other when there are two closely related concepts. The achieved preciseness of manual indexing has the drawback of consistency of relying on human perception and interpretation, as has been reported in various studies (Cooper, 1969), (Sievert and Andrews, 1991), (Chan, 1989).

Because human judgment requires manpower, the number of index terms assigned to a record is normally a matter of policy driven by cost (Plaunt and Norgard, 1998), also including factors like storage space, computing power for the search process, and search performance results after human indexing has assigned some number of authorized index terms to each document as it enters a particular system.

2.1.1.3 Indexing with Controlled Vocabulary and Thesauri

Using controlled vocabulary is a common practice of human indexing. After identifying potential index terms to be assigned to a document, an indexer selects final index terms by consulting a list of controlled vocabularies that have either been constructed by a group of professionals and experts such as medical researchers and practitioners or by an information provider and organizer such as library. For example, Medical Subject Headings (MeSH) has been created by National Institutions of Health (NIH) (Lindberg et al., 1993), (McCray and Nelson, 1995). Library of Congress Subject Headings (LCSH) have been generated by Library of Congress.

In addition to defining controlled vocabularies, semantic relationships between such vocabularies are constructed manually. Semantic relationships commonly include broader term, narrower term, synonym, related to, used for, and uses. In order to achieve the completeness of characterizing a document, some information provider utilizes semantic relationships in thesaurus to automatically bring in related terms of those assigned index terms. This practice can easily expand five assigned index terms to twenty or thirty index terms for a document. Petroleum Abstracts is one example (Finnegan, 1991), (Bailey, 1994). They carefully calibrate the induced terms in the bibliography record.

The use of controlled vocabulary for indexing reduces the wording variation chosen by different indexers. That is, as long as a concept is deemed to be important for a document, a precise index term will be assigned to that document, thereby eliminating problems from morphological variations such as *lung tumor* and *tumor of the lung* (Jacquemin and Tzoukermann, 1999).

The main drawback of using controlled vocabulary is the limitation on assigning new terms to reflect new concepts. Usually there is a delay incorporating new terms into controlled vocabulary as well as thesauri.

2.1.1.4 Free-text Indexing

Free-text indexing relies totally on words or phrases found in a document. The technique can be applied to full-length documents, abstracts, titles, and combinations of them. Nowadays, it currently is usually performed with automatic methods, is also used manually in some bibliographic systems such as INSPEC. When it is employed manually, term phrases (commonly of two or three words) are carefully selected from sentences. Occasionally, four-word or five-word term phrases may be used. Whereas manual effort limits the number of free-text indexes to be associated with a record, the automatic practice commonly demonstrated by popular web search engines uses all non-stop words to index each record. Term-phrase search may be supported by the adjacency of words in records or using some term formation technique to create term phrases from text as indexes (Salton, 1988), (Chen and Lynch, 1992).

The main advantage of free-text indexing is that it allows a document to speak for itself. It captures authors' wording which generally very up-to-date. In the case of term phrases, indexes also are retained in their natural language format, mainly as noun phrases that may represent precise concepts. In addition, automatic free-text indexing provides the most complete index coverage to all records in a information system.

Nonetheless, the completeness of free-text indexing is one dimensional - exact words or phrases inside each document. Two documents with similar content but different vocabularies will have different free-text indexes. Two documents on the same topic may have only a small portion of their manual indexes in common. This leads to diminished recall value when relevance is measured beyond the syntactic level.

On the other hand, the massive volume of free-text indexes may weaken the precision of retrieval. Some words and even term phrases may have multiple meanings. They may exist in two unrelated documents. A retrieval process with many general terms brings unrelated records together to the detriment of precision and the inconvenience of an overwhelming number of retrieved records.

2.1.2 Characterizing Global Knowledge in Document Collections

While analyzing a single document is undertaken to reveal the knowledge it contains by enlisting embedded concepts, analyzing a collection of documents is done to discover the overall but hidden knowledge under the conglomerated effect. The characteristics of the knowledge of a collection identify its relevance, completeness, and proper usage. Because the cumulated information is so voluminous, the characterizing process involves intensive resources and expertise, but the return on investment fortunately is an understandable summary of an ever-growing information in organized scheme.

The following three sub-sections describe different techniques and resources that can be used to characterize document collections. The goal of the first two of these (classification schemes and knowledge discovery) is to explicitly reveal the characteristics of a particular document collection in its entirety. The main difference between them is that classification schemes rely solely on manual effort while knowledge discovery relies heavily on automatic computational power. On the contrary, the third technique converts the global characteristics of a document collection into functions that take the form of ontology and inferencing rules.

2.1.2.1 Classification Schemes and Categorization

Classification or classification systems have a long history of being used organize large amounts of information in a managable manner. Library systems use the

Dewey Decimal System or the *Library of Congress Classification System* to organize their collections physically and conceptually (Kao, 1995), (Kohl, 1986). The *Association for Computing Machinery* (ACM) uses its own classification scheme to organize subject areas of its interest over the past 50 plus years. The *Yahoo!* directory is one of the attempts that have been made to organize vast amount of Internet information.

Classification or categorization is the putting together of like things into their categories. For purposes of explication, we may consider a category system as having both vertical and horizontal dimemsions (Rosch, 1978). The vertical dimension concerns the level of abstraction of the category system while the horizontal dimension focuses on the segmentation of categories at the same level of abstraction. By implication, a category system having these two dimensions is intrinsically hierarchical. That is, the natural knowledge representation of a category system is a tree structure, one of the most readily comprehensible data structures to human beings. Tree structure is commonly used and seen in tables of content, family trees, and organizational charts.

A classification scheme is intended to provide coverage of all known knowledge. Since the 19th century, several classificiation schemes such as Dewey Decimal Classification, the Library of Congress Classification, Universal Decimal Classification, and Reader Interest Classification have been used in libraries to include all kinds of knowledge (Miller and Terwillegar, 1990). However, there has never been a single scheme upon which everyone agrees. In practice, each classification scheme defines its own set of categories at different levels of abstraction (Rosch, 1978). In addition, each scheme has its own choice of vocabulary. The bottom line is not which scheme is correct but which has the flexibility to extend its coverage to include new categories and sub-categories (Miller and Terwillegar, 1990).

On the contrary, categorizing a smaller information set calls for providing maximum information about the underlying collection while requiring the least cognitive effort to obtain given information within it (Rosch, 1978). In the other words, the result of categorization covers only all knowledge found inside a given information collection. Many disciplines and communities have their own category systems, such as ACM's Classfication Scheme and Compendex's Engineering Classification Scheme that offer grand coverage only to their sponsoring communities.

Traditionally, the making of category systems, like that of various classfication schemes, is very labor-intensive. Even though the goal of having a good classification scheme is to minimize cognitive effort needed to distinguish different categories, the massive quantity of defined categories requires appropriate human learning and comprehension in order to locate a classified piece of information. Fortunately, the widely known hierarchical structure provides a natural divide-and-conquer approach to directing users' attention.

In general, physical items like books go into a category in a classification scheme while the intellectual items like topics in a book go into several categories. Such cross listing or cross referencing ensures completeness of coverage by a defined category. In addition, items in the same category provide direct results to a search for similarity to a particular item, simply because only like-items go into the same category.

Although there is no "true" classification scheme or categorization, Wynar (Wynar, 1985) identifies a few criteria for a successful classification scheme. For existing information, a classification system must be inclusive as well as comprehensive. For new information, the system must be flexible and expansible. In all cases, the system must employ terminology that is clear and descriptive, with consistent meaning for both the user and the classifier. This set of criteria resembles the heuristics used by most classification schemes, which may in fact serve as evaluation criteria for automatic classification and categorization methods described in next sub-section.

Classification systems are commonly used in digital libraries and document management systems. However, classfication systems are pragmatically designed for optimum ease of human access. They do not aim at a semantically clear formal model (Abecker et al., 1998).

2.1.2.2 Knowledge Discovery from Large Databases

Similarity between documents can be computed based on the vector space model (Salton et al., 1975), (Salton and Yang, 1973). This similarity computation forms the core technique used to perform various methods of knowledge discovery from large textual databases. Chen (Chen, 1995a) discusses the use of neural networks, symbolic learning, and genetic algorithms to perform automatic characterization on large document collections. Once each document is converted into an index vector, various similarity functions such as Jaccard and Cosine coefficients used by different algorithms will compute the similarity score between a pair of documents (Salton, 1988). The main difference among various algorithms is the clustering method of lumping all similarity scores together to form *meaningful* clusters of documents.

Many clustering or categorization techniques have been applied to the field of information retrieval. These techniques include classical graph data structure and algorithms such as Ward's algorithm (Even, 1979), (Ward, 1963), statistical algorithms such as multi-dimension scaling (MDS) and discriminant analysis (Jain and Dubes, 1988), (McLachlan, 1992), symbolic learning algorithms such as ID3 and AQ15 (Quinlan, 1983), (Michalski et al., 1986), and neural network algorithms such as Kohonen's self-organizing map (SOM) and Hopfield Net (Kohonen, 1995), (Hopfield, 1982), (Lippmann, 1987). They are all capable of making some meaningful categories from document sets ranging from several hundreds to several thousands. However, large scale attempts are limited because of the scalability issue that is related to size of data set and the demanding need for computational resources such as processing power and memory. In the mid-90s, the field of information science started to make use of high performance computing resources to analyze very large textual data collection.

2.1.2.3 Knowledge Bases, Inferencing, and Ontology

Classification scheme and categorization are special cases of knowledge bases. Their knowledge is tailored to provide grand coverage of corresponding document collections. In a broader sense, knowledge bases cover knowledege in different domains or subject areas. Under the notion of knowledge discovery, at the risk of being overly recursive, facilitating knowledge mining requires providing knowledge about knowledge (Rouse et al., 1998). Ontologies and data models are used in knowledge-based and database systems, respectively, to specify the basic assumptions that went into the system's conceptualization (Gruber, 1993).

The idea behind knowledge base and database coupling is first to build a knowledge base to reflect the database and then to access the database through the use of the knowledge base. However, there is no established form of knowledge base. One explanation is that it is not necessary for such derived knowledge to be made explicitly available to users. Since such a knowledge base is a component or function of an integrated system, knowledge may freely appear in many kinds of knowledge representations such as semantic net, production rules, frames, and ontologies (Rich and Knight, 1991). Some knowledge is manually crafted. Examples are the metathesaurus of the Unified Medical Language System (UMLS) project, a thesaurus created by medical experts and practitioners (Lindberg et al., 1993), (Rada and Martin, 1987), (Humphreys and Lindberg, 1989); MYCIN, an expert system engineered by intensive knowledge acquisition in the medical field (Shortliffe, 1976); CYC, an ontology crafted by effort in gathering global and common sense knowledge (Lenat and Guha, 1990), (Lenat et al., 1990). Other knowledge is derived through automatic machine learning methods such as ID3 and Kohonen self-organizing map.

In order to make knowledge bases work with a target document collection, different inferencing mechanisms are employed according to different knowledge representations. In a production rule system, an inference mechanism relies on the control strategy built inside the recognize-act-cycle control module. Production rules, in the form of predicate logic, are brought into the control module based on some induced conditions. Conflict resolution strategies and even heuristics are utilized to decide the next action. The control module also includes a backtracking mechanism to counter any wrong decision made in the recognize-act-cycle.

In semantic or hybrid networks, different search algorithms can be used as inferencing mechanisms to traverse a *network of knowledge*. The simplest form of such networks uses *nodes* to represent concepts and links to represent semantic or probabilistic relationships between each pair of concepts. Built upon this graphic representation, a frame or script is used to extend the content of each node - constraints, exceptions, time and place information (Lehmann, 1992).

2.1.3 Revealing Knowledge in Neighborhood

While characterizing documents shows what they are and how they are related, concepts - basic elements in documents - yield another level of analysis. The following sub-sections show how a list of concepts is defined and how concepts can be used to reveal some "neighborhood" knowledge.

2.1.3.1 Syntactic Mapping: Index List

Syntactic mapping provides the capability to display an index list used in a document collection. For all information retrieval systems, such index lists are given because they are the list of keys in the inverted index to documents.

A common practice is to list a set of terms existing in a system in alphabetical order based on term fragments entered by a user (Kowalski, 1997). The user can then examine the terms on both sides of the neighborhood. This browsing process allows a user to select exact terms for the searching process. It also is feasible to type in some *approximate* pattern with *errors* to retrieve a list of system terms (Wu and Manber, 1992b), (Wu and Manber, 1992a). In addition, it is a useful for some *advanced* searches to list the number of occurrences (document frequency) to decide what terms should be used in a query.

A list of searchable terms is a given component of an information retrieval system. Issues in policy, design, and implementation require inclusion of such list as a reminder service or a dictionary service. In fact, having an index list to aid searchers is a long-time practice in books and on-line help information and offers the least expensive way to reveal some potential search terms in a general neighborhood.

2.1.3.2 Keyword Mapping: Controlled Vocabulary

Controlled vocabularies are created manually by information producers like INSPEC or organizers like the Library of Congress. In addition, controlled vocabularies can be used to assign subject headings to documents, or to support effective search, since controlled vocabulary terms can be precisely submitted as queries. Unless searchers are indexers, however, controlled vocabularies are often foreign to most searchers.

Library of Congress publishes the Library of Congress Subject Headings (LCSH) to cover vocubularies in all subject areas. LCSH is the most comprehensive list of subject headings in the world today. It provides an alphabetical list of subject headings, with cross-references and subdivisions verified by the Library of Congress. As of Spring 1999, it had a total of 234,000 headings and references (Library of Congress. Subject Cataloging Division., 1998). The *USE* relationship directs users to use proper keywords while the *used for* (UF) relationship reminds users what a proper keyword means. However, searchers need to go through a manual browsing process to locate the controlled vocabularies.

In order to simplify the use of such controlled vocabularies, some systems try to map searcher's keywords to an *internal* vocabulary list. For example, the Ovid search engine for databases like CancerLit and Medline from Ovid Tecnologies, Inc. uses a statistical analysis to map which subject headings tend to occur in documents containing a searcher's free text query. Ovid utilizes the database producer's online vocabulary or a thesaurus such as UMLS. Knowledge bases and rules can be used to help users locate appropriate controlled vocabularies from their own search terms (Shoval, 1985).

2.1.3.3 Semantic Mapping: Man-made Thesauri

In addition to providing a controlled vocabulary, LCSH is a thesaurus associating concepts through a set of semantic relationships: narrower term (NT), broader term (BT), and related term (RT) (Library of Congress. Subject Cataloging Division., 1998). Instead of focusing on the syntactic variation of terms, a thesaurus gives the semantic variation of potential search terms. In the other words, a thesaurus brings together in different forms terms for the same or similar concept. Another general domain thesaurus is Roget's Thesaurus (Roget, 1962). Different communities create their own thesauri in some specific domain areas. For example, INSPEC thesaurus is for the domains of physics, computer science and engieering (Institution of Electrical Engineers, 1993); GeoRef thesaurus is for geosciences (Goodman, 1997).

Because it requires a tremendous effort to create a thesaurus, the UMLS project tries to utilize some abstraction analysis to create a general "lightweight", but comprehensive, *Semantic Network* to cover semantic relationships between medical concepts (McCray and Hole, 1990), (Lindberg et al., 1993), (McCray and Nelson, 1995). The Semantic Network contains 132 semantic types and 53 semantic relationships. Semantic types are abstracted from all UMLS terms. Semantic relationships are created for these sets of semantic types. Although the Semantic Network covers all semantic relationship of UMLS terms through semantic types, a semantic relationship for two particular UMLS terms under two corresponding semantic types associated with a semantic relationship may not make sense.

2.1.3.4 Co-occurred Mapping: Automatic Thesauri

Co-occurrence analysis is a statistical algorithm to calculate a co-occurring weight between a pair of terms in a document collection (Salton, 1975). The creation of a list of co-occurring terms is solely dependent on what terms are being *indexed* from all documents. It also has the capability to associate two opposite concepts. For example, the sentence, "A is not B", has two negatively associated terms (A and B). However, the probability of having many such sentences in a large document collection is very small and chances of such a co-occurrence relationship are insignificant.

Strictly speaking, automatically computed co-occurrence relationships can hardly be considered semantic relationships. Nonetheless, the aggregate effect of such co-occurrence information does give a tremendous contextual information with respect to the underlying document collection. For example, an acronym usually co-occurs with its full name and this context gives insight into a particular concept. As in reading - contextual information helps in understanding an article. Co-occurence analysis can be used to compute both symmetrical and asymmetrical co-occurred weights between pairs of terms. The choice of asymmetrical co-occurrence has an advantage to mnemonic human thinking process (Chen and Lynch, 1992).

2.2 Dynamic Nature of User Information Need

When searching, what do users want? Given the most common utilization of web search engines, the *obvious* answer is a list of documents (URLs). But is it a *real* answer? This section does not intend to find the *real* answer. Instead, it will look at the dynamic nature of user need in two aspects - expressing user need and perceiving knowledge. To a certain extent, using information retrieval systems is like communicating with another person. A user needs to express his or her need to a system and then the system will return *some information* to the user. The first of the following two sub-sections discusses user need. The second one examines how users deal with the information returned.

2.2.1 Expressing User Need

The goal of information retrieval is to return as many as possible relevant objects in response to a given information need. This simple goal has two types of complexity: relevance and information need. The question of relevance will be discussed in the section on *perceiving knowledge*.

First, we will discuss what information need is and how it is related to information retrieval systems. We will then look at three different phenomena involved in users' difficulties in expressing their needs. Finally, we will examine whether we can borrow some techniques from information providers to help users to express their information need.

2.2.1.1 Information Need

Information need is merely the reflection of what a searcher wants at a given time. Cooper (Cooper, 1971) describes information need as a searcher's psychological state which is not directly observable or symbolized. Bates (Bates, 1990) also argues that a mismatch between information need and what information systems currently provide is partly due to the static nature of information systems previously discussed, which is complicated by the dynamic nature of information need.

In attempts to overcome the mismatch and more effectively capture the information need, there has been much research in the area of human-computer interaction. In a system-oriented review of over 50 different search-interfaces, Vickery and Vickery (Vickery and Vickery, 1993) conclude that current interfaces may be over-elaborated or over-engineered by being so structured that the information displays and the ordering constraints for entering query are predetermined and unchangeable. They suggest that people might benefit from simple systems that allow a flexible and revealing dialogue between the system and user. In the research on information workspaces and visualization, Rao et al. (Rao et al., 1992) states that even thought searchers may want to interleave access operations and track their progress, current information systems are weak in their support for this process. Their findings simply show that no predefined interface to any information system can fit the diversity of information need.

Regarding difficulties in the development of knowledge based systems to aid the search process, Bates (Bates, 1990) argues that the goal should not be to replace the searcher with a knowledge based system, but rather to design the interface to support the strategic, opportunistic behavior of searchers.

In contrast to a predefined interface, Hendry and Harper (Hendry and Harper, 1997) create a loose and informal information-seeking environment in which searchers can *freely* express their information needs. Such an environment allows a searcher to associate query and result graphically and spatially in a work space. The ability to spatially arrange information, called *secondary notation* (Hendry and Harper, 1997), can help the searcher to comprehend the search process as well as to guide the execution of search activity. However, their findings show that lack of expertise in using such open information-seeking environment is a barrier to expressing a searcher's information need, indicating that user interface is not the sole factor in or solution to elicitate user's information need.

2.2.1.2 Indeterminism

Indeterminism is another phenomenon related to expressing information need (Chen et al., 1994a), (Blair, 1986), (Bates, 1986). Three factors may contribute to searchers' inability to express precisely what they want: system, searcher, and process. The system factor involves the variety in index terms assigned to documents. Lacking of indexing consistency can be further broken down into three areas: inter-indexer inconsistency, intra-indexer inconsistency, and inter-system inconsistency. The inter-indexer inconsistency comes from the observation that different indexers are likely to assign different index terms to the same document. The intra-indexer consistency derives from recognition that an indexer may use different index terms for the same document at different times (Blair, 1986), (Bates, 1986). The inter-system inconsistency comes from the fact that the collections of documents in many information retrieval systems overlap (Chen et al., 1994a). Furthermore, different systems use different indexing policies and controlled vocabularies (Barber et al., 1988).

The second factor for indeterminism arises from the variety in search terms used by searchers (Blair, 1986), (Bates, 1986). Like indexers, different searchers use different search terms to express the same information need, either because of unfamiliarity with the subject area or differences in experience and training.

The process factor is the subtlety and complexity of the search process (Bates, 1986), which is heavily dictated by the search mechanisms provided by different systems. Variations of search mechanism range from query term entering methods, choice of filters, complexity of boolean formulation, and user interface designs.

2.2.1.3 Opportunism

At the heart of search tactics and strategies, a search process involves planning, backtracking, and comparison (Bates, 1979a), (Bates, 1979b). In this connection, drawing from the research in programming environments, Visser (Visser, 1994) finds that support for basic cognitive tasks such as planning, backtracking, and reformulating is considered essential because it allows people to work the way they want to - opportunistically.

Carmel et al. (Carmel et al., 1992) performed a cognitive study which showed that opportunism appears in two out of three kinds of browsing strategies in a hypertext environment: review-browse and scan-browse. The review-browse strategy is to scan and review interesting information in the presence of transient browse goals that represent changing tasks. The scan-browse strategy is to scan for interesting information without review. Marchionini and Shneiderman (Marchionini and Shneiderman, 1988) also define browse as an exploratory, information seeking strategy appropriate for ill-defined problems and for exploring new task domains. As long as indeterminism in expressing information need exists, opportunism stemming from any hint will be a great help to users.

2.2.1.4 Vocabulary Problem

A searcher's experience is phenomenologically different from an indexer's experience (Bates, 1998), giving rise to potential choice of different vocabularies to describe the same object and creating a matching problem between terms provided by indexers and searchers. Even if well trained in an indexing scheme, different indexers might assign different index terms for a given document (Bates, 1986). In another study, Furnas et al. demonstrated that in a setting allowing spontaneous word choice for objects in five domains, two well-trained indexers favored the *same* term with less than 20 percent probability (Furnas et al., 1987). The probability might be even worse for searchers having different levels of domain expertise and system knowledge (Furnas et al., 1983), (Furnas et al., 1987), (Chen, 1994).

As for expressing a query, the dynamic of vocabulary differences also happens between searchers and systems. Vocabularies in systems are, of course, mainly determined by authors, indexers, and information providers, but, in addition, vocabularies in systems are static in terms of their accountability to documents. Every search process from the same or different searchers creates a different vocabulary matching situation to a system so unless searchers know the vocabulary used in the system, the variety of language presents a serious barrier to express query.

2.2.1.5 Recognition with Contextual Information

To alleviate the difficulties in expressing user need, techniques and heuristics can be borrowed from the manual process of generating thesauri. After using some automatic techniques to generate a set of potential terms from a defined domain area, the main task is to select a set of *useful* terms to be included in the new thesaurus (Kowalski, 1997). To aid the selection process, text concordances from documents that cover the domain area are used. A text concordance is an alphabetical listing of terms from a set of documents along with their frequency of occurrence and *references* to documents in which they are found (Salton, 1988). A text concordance is also known as *Key Word Out of Context* (KWOC) (Kowalski, 1997). Related to KWOC, *Key Word In Context* (KWIC) shows meaningful contextual information about a term by listing two fragments of terms before and after the term in the original sentence or document (Luhn, 1960), (Salton, 1988), (Kowalski, 1997), (Wynar, 1985). KWIC is useful in determining the meaning of homographs. For example, the term “chip” could be *wood chip* or *memory chip*. With KWIC, the editor of the thesaurus can read the sentence fragment associated with the term to understand its context and then determine its meaning. However, the KWIC tool is impractical in a large database where the listing of sentence fragments is very long. Nonetheless, contextual information may help a user develop a better understanding of a term’s meaning.

The technique of relevance feedback is similar to KWIC in delivering contextual information to user. The relevance feedback concept is that the new query should be based on an old query modified to increase the weight of terms in relevant documents and decrease the weight of terms that are in non-relevant documents (Rocchio, 1971), (Salton and Buckley, 1990). Under relevant feedback, contextual information is given by a set of retrieved documents instead of a set of sentence fragments. Users need to identify whether each document is relevant to what they want. The system will then use the information of relevance to modify the users’ query. In practice, users often skip the most important step - providing the relevance feedback information.

2.2.2 Perceiving Knowledge

How humans perceive knowledge has been studied by many disciplines such as Philosophy, Psychology, Linguistics, Cognitive Science, Behaviorial Science, etc.

Various competing and sometimes even complementary theories and models have arisen to explorer how a human mind works (O'Brien, 1998).

Indeed, the fundamental is a question about what knowledge is. The same goes for human perception. Our objective here is not to attempt to answer these two philosophical questions. Rather, we would like to use them to look at these issues in information retrieval, or search process in general. What is the user's perspective of knowledge? How does a user perceive retrieved or *derived* knowledge? In the field of information retrieval, many tools and techniques to assist humans to obtain information have been produced from theories and experiments. Some show promising results in experimental settings. However, in practice, they may have made a limited contribution, as witnessed by the observation that popular web search engines often have reverted to technologies developed in 1960s. What is the problem?

In the discussion of knowledge in intelligent systems, Minsky (Minsky, 1968) declares that heuristic programs are able to solve much harder problems than self-organizing systems because heuristic programs are given enough specific factual knowledge about particular problems. They do not have to start from an unstructured basis to evolve everything they will need, given that the requisite *knowledge* is suitably represented. How much of this discussion can be applied to information seekers? Since Minsky attempts to build intelligent machines which model human intelligence, a parallel discussion of analysis of users seems justified. Then, how can we represent strategic knowledge? And, how can we so present strategic knowledge to users to allow them to perceive retrieved information and possibly obtain more knowledge?

Meaning requires time-consuming thought, and the pace of modern life works against affording us time to think: the time to translate data into information and information into knowledge (Wurman, 1989). We have a limited capacity to transmit and process information, so we must distort the flow by being selective. The more there is to select from, in a given time frame, the more distortion must occur. Basically the technology of the production, storage, and transmission of information has outstripped the human strategies available to cope with this great volume of global data (Reeves, 1996). The ability to link relevant but fragmented information into some coherent whole allows for new knowledge. Having an adequate context or contextual knowledge base greatly enhances the possibility of assimilation of new information within pre-existing frames.

This literview review has been restricted to some important areas related to how humans perceive knowledge from information retrieval systems, in particular, how users comprehend knowledge embedded in systems, text documents, and search processes.

2.2.2.1 Computing Relevance?

If an information need can be expressed clearly, can *relevance* be computed? Searchers can consistently and easily determine the relevance of an information object with respect to their information need without being able to enunciate the criteria they use to do so (Blair, 1990). Cooper (Cooper, 1971) also defined the determination of relevance of an information object as a subjective process which reflects the psychological state of the information need. In order to have a better understanding of relevance, Cooper (Cooper, 1971) makes a distinction between so called *logical relevance* and *utility*. A given object is logically relevant to an information need if the object is topically related to the need. On the other hand, utility is purely a pragmatic notion: Is the object *useful* to the searcher? It is obvious that utility is solely determined by the searcher and cannot be computed by any information system. In an attempt to compare different information retrieval models, Bruza and Huibers (Bruza and Huibers, 1996) propose *aboutness*, which is level of topical relatedness between an object and a request computed by an information retrieval system. That is, the most that a system is able to do is to retrieve related but not relevant objects. The relevance only can be identified after the searcher examines the retrieved result. In the other words, the searcher needs to go through the perceiving process on the retrieved materials first and then evaluates their relevance.

Nonetheless, most information retrieval *sessions* stop working when they return a list of retrieved document objects. Such sessions happen in several situations. Observing the information retrieval in the web reveals that a common search model is *query-trial-and-error*, in which the only activity that the searcher can perform is to try giving different query terms to prompt a system to retrieve a small set of top ranked *related* objects. Searchers may not use some systems with sophisticated features such as relevant feedback, refinement, and knowledge base, because of time constraints or a non-intuitive user interface. In addition, the content and format of knowledge employed by the producers of knowledge bases are seldom of interest to the vast majority of searchers (Rouse et al., 1998). Obviously, such a system is along way from fulfilling the information need.

More than thirty years ago Minsky (Minsky, 1968) summarized the problem of defining relevance to users when dealing with large amount of information as follows:

> ... it is hard to find a knowledge-classifying system that works well for many different kinds of problems; it requires immense effort to build a plausible thesaurus that works even within one field. Furthermore, any particular retrieval structure is liable to entail commitments making it difficult to incorporate concepts that appear after the original structure is assembled (Minsky, 1968).

Since then, different research groups and information providers have spent tremendous resources to build domain specific thesauri. In addition to thesauri, different *original knowledge structures* have been created to give comprehensive coverage in different domains. However, the issue of having some *helpful retrieval structure* remains unsolved. Minsky states that:

> The problem-solving abilities of a highly intelligent person lies partly in his superior heuristics for managing his knowledge structure and partly in the structure itself; these are probably somewhat inseparable. In any case, there is no reason to suppose that you can be intelligent except through the use of an adquate, *particular* knowledge or model structure (Minsky, 1968).

Different knowledge structures have been created to aid searchers to *perceive* the relevance of retrieved objects. The following three sections describe them based on the existence (or non-existence) of structure and context of each method and thereby complete the second half of the retrieval journey.

2.2.2.2 Structureless and Contextless: Document List

List structure has long been recognized as a commonly comprehensible structure. The inverted index - the core technology in information retrieval - is a list structure that stores all documents having the same term index. Although the inverted index is an internal data structure in a system, most user interfaces directly present only a document list as a search result. Of course, a majority of these present only a partial list of documents due to the tremandous volume of information in different search engines. It is also true that searchers usually examine only a small portion of the retrieved document list.

From the perspective of helping searchers to perceive retrieved information, a document list provides hardly any structural or contextual aid. Even though documents are presented in some kind of ranked order based on the vector space model, searchers have to evaluate each document independently for its relevance. The list structure merely provides an ease-of-use access method. Besides, the listing does not carry any contextual information related to documents. The only contextual information comes from the query.

2.2.2.3 Structural but Contextless: Dynamic Clustering

Ad-hoc clustering or categorization is often used for converging vast information in a considerably shorter time. The Northern Light web search engine categories search results into some pre-defined and ad-hoc categories as a summary tool (Schwartz, 1998). The convergence process of electronic meeting also utilizes categorization techniques to define categories of concerns and discussion in such meeting (Chen et al., 1994b). These additional analyzed results are usually presented prior to or alongside document lists.

From the searchers' point of view, dynamic clustering is an extra knowledge structure superimposed on the underlying retrieval materials. The searchers benefit from the structure to direct themselves to certain *interesting* categories and then examine documents in those areas. However, the contextual information comes after the searchers examine the dynamic clustering structure. That is, the searchers can expect only a clustering structure but without any hint of what the content would be.

2.2.2.4 Sturctural and Contextual: Path to the Knowledge

According to the mental-logic model on basic inference schemas (Braine and O'Brien, 1998), Lea et al. in the field of psychology performed a series of studies on how humans draw inferences from text comprehension (Lea et al., 1990). One of the studies was on a *Recognition Task* in which subjects were asked to indicate whether the information contained in three newly presented sentences had been presented *explicitly* in the text (story) they had just read, or whether they had to infer that information.

Here is an example extracted from Lea's work (Lea, 1998). The story is:

1. The Borofskys were planning a dinner party.
2. "Alice and Sarah are vegetarians," Mrs. Borofsky said, "so if we invite either one of them, we can't serve meat."
3. "Well, if we invite Harry, we have to invite Alice," Mr. Borofsky said.
4. "And if we invite George, we have to invite Sarah."
5. "We already decided to invite either Harry or George, or maybe both of them," said Mrs. Borosky.

Among the three sentences for the *Recognition Task*, the following one is most interest to the discussion of users' inferencing process with respect to information retrieval:

- In their discussion, the Borofskys concluded that they had to invite Alice or Sarah (or both).

Characteristic of this sentence is that it contains a logical inference that the mental-logic model predicts readers would make while reading the story. Lea et al.'s results indicated that sixty-nine percent of the subjects thought that the inferences had been presented explicitly in the text (story), a result that provides strong evidence that people often do not realize they are making inferences in text comprehension.

In addition, in many discourse situations a distant premise is simply a fact stored in long-term memory, not information that was presented earlier in the text or conversation. The following example illustrates how distant premise information can be retrieved from the participants' world knowledge, as well as from their memory of earlier parts of the discourse (Lea, 1998):

- Bob was asking Barb about her new personal computer.
- "What did you decide about which type of computer to get?"
- "Well," said Barb foolishly, "in the end I decided not to get the Mac because they'll probably be out of business before I fill out the warranty card."
- *Inference:* Barb got an IBM-compatible machine.

In this example, the distant premise is world knowledge stored in both Barb's and Bob's long-term memory: Personal computers are either Mac or IBM-compatible machines. This is the type of distant premise used for making inferences in text comprehension.

In the information retrieval process, a parallel analysis indicates that the distant premise can be captured by a system, even though the knowledge is not complete. Users choose distant premises, both from the computer system and their own knowledge, in order to make inferences to help them comprehend the search space, and possibly the search result. From a philosophical point of view, Lipton (Lipton, 1991) argues that inferences lead to the best possible explanation. Inferences may occur recursively until a comprehensible premise is reached and a partially defined search space can be extended to a comprehensible one through inference. In the other words, the premises coming from inference help shape the scope of the search space. In addition, distant premises give clues that help users comprehend and analyze the underlying search space, which is the document space with respect to the information retrieval.

From the psychological point of view, the activities of knowledge revision can give rise to aggregates for concept formation (Wrobel, 1994). Wrobel (Wrobel, 1994) describes knowledge revision as a necessary activity for knowledge acquisition due to several factors like changing world and application domains, sloppy modeling at the initial stage of knowledge acquisition, and selection bias and incremental learning.

From a user study of inexperienced searchers' learning to use online search engines, a positive effect was found to be the direct result of increased searching success as greater knowledge of vocabulary and search strategy was acquired (Meghabghab, 1995). Knowledge of a vocabulary gives context to the search process while search strategy gives structure to the process.

CHAPTER 3

RESEARCH QUESTIONS AND METHODOLOGIES

3.1 Research Questions

Chapter 2 discussed various information analysis methods for building information retrieval systems these range from indexing to classification and from manual to automatic processing. The purpose of such systems is to provide knowledge and mechanisms that allow users to access underlying documents relevant to their individual information needs. However, the information analysis process does not incorporate information need, making the knowledge embedded in systems static and unable to encounter ever-changing user information needs. In addition, because systems usually employ specific search knowledge and system knowledge and possess specific domain knowledge, they raise extra barriers to user access to needed information.

The main difference between today's online information retrieval process and the traditions maintained through hundreds of years of history in library science and information science is that they rarely involve information consultants such as librarians, and information consultants can help users articulate their specific information needs and eliminate users' having to know the specific searching mechanism and system knowledge of a particular information retrieval system. These changes are directly from the availability of Internet technology and the use of the World Wide Web. During its 40-year history, information retrieval research has implied that the advancement in information analysis and retrieval mechanisms will overcome the deficit of not having help from information consultants. Nonetheless, as discussed in Chapter 2, difficulties in expressing information needs to information retrieval systems and perceiving knowledge contained in the retrieved results persist. Information retrieval process is still a time-consuming and labor-intensive effort.

Information seeking is an intrinsically cognitive process undertaken to fulfill information needs raised by users. It is a semantics-bearing and interactive process in which we can simply ask a colleague for some information or try to retrieve it from some information retrieval system. My research focus is a machine's semantic processing, rather than its mechanical operation, of information retrieval.

The board question is, can knowledge sources be used to help users express their information needs? How to use and how to generate such knowledge sources are the two focuses of this dissertation. The specific question are as follows. Would the *automatic concept exploration process* be able to help users identify more relevant concepts? Would such a process be able to perform more efficient exploration of a concept space than the conventional manual browsing method? If so, which algorithmic methods - symbolic-based branch-and-bound or neural network-based Hopfield net algorithm - is better in terms of gathering relevant concepts from knowledge sources? Would the *concept space consultation* process provide a *semantic medium* to reduce the *cognitive demand* from users in terms of *elaborating information needs*? Would the *concept exploration process* be able to help users find more relevant documents? With regard to computing scalability, would the technique of computer generation of concept spaces be applicable to very large textual databases? With regard to domain specific knowledge scalability, would concept space generation by technology create satisfactory domain-specific concept associations from corresponding textual databases? How does the quality of concept associations in concept space generated from very large textual databases compare with that of a man-made domain-specific thesaurus?

3.2 Methodologies

The research problems being looked at involved an effort to develop a semantic component which is capable of resembling semantics used by humans and to create a systematic process to make use of the semantic component to aid the information retrieval process. Two methodologies which seemed most appropriate for this research were the systems development approach (Nunamaker et al., 1991) and experimental design.

In the systems development methodology, a problem is first identified and then an automatic or system-based solution is analyzed, designed and implemented, typically using one of the systems analysis and design paradigms such as object-oriented analysis and design. The resulting system solution is then tested in a laboratory or real-world setting.

In order to validate the usefulness of a system or automatic technique, its results must be compared against existing systems, existing standards or existing techniques. The experimental design methodology provides guidelines such as sample size and confidence level in conducting experiments to collect data for measuring against the underlying objective. In information retrieval, precision and recall are two measures commonly used to evaluate the performance of different information retrieval techniques. However, several studies reveals problems with both (Raghavan et al., 1989), (Tague-Sutcliffe, 1992), (Hersh, 1994). The first problem is that the knowledge of the proper estimation of relevancy of all the documents in a large collection is unavailable, which implies that recall cannot be estimated precisely. Relative, instead of absolute, measures must be used for large collections. The second problem is that recall and precision originally were designed to measure the effectiveness over a set of queries processed in batch mode. However, since almost all retrieval systems are processed interactively, the time factor should be considered in determining effectiveness.

3.3 Dissertation Plan

My dissertation aims to investigate the use of semantic-enabled information technologies to help users seek information. In order to facilitate understanding of the research and its context regarding to various relevant information technologies, I present a framework to guide the research and discussion. Its two core semantic-enabled information technologies are *concept consultation system* and *concept association analysis* (as one of the knowledge discovery methods).

3.3.1 A Framework for Semantic-enabled Information Technologies

Based on past research and our experience with various knowledge discovery methods, I developed a framework to depict the different entities involved in knowledge discovery and their intertwining relationships.

The main entities involved in this framework include the underlying *databases* from which knowledge is acquired, *knowledge discovery* methods, the *knowledge bases* discovered by the algorithms and those imported from other domain-specific sources, *information retrieval systems* for accessing documents, and *concept consultation systems* for users. The definitions of databases and knowledge bases are sometimes blurred because of their historical roots in different disciplines such as artificial intelligence, database management systems, and information retrieval. In the context of this research, *databases* refer to online repositories of basic facts about objects and events in the world, e.g., employee files, transaction records, bibliographic records, etc. and *knowledge bases* are online repositories of high-level, abstract human knowledge represented in terms of heuristics, inferencing rules, problem solving strategies, networks of inter-related concepts (concept space), and so on. These entities and their relationships are shown schematically in Figure 3.1 and discussed in the context of earlier research.

- *Databases:*

  Domain-specific databases that capture information or data of relevance to users' applications can be taken as major sources of knowledge. Since the '80s, many major corporations and government agencies have used databases of drug side effects, retail shopping patterns, tax and welfare frauds, frequent flyer patterns, and so on to identify application-specific knowledge. Frawley et al. (Frawley et al., 1991) presents a good overview of databases used in knowledge discovery.

  The enormous sizes of real-life databases, which have frequently prevented human beings from conducting labor-intensive analysis, and the availability of unused computing cycles in many institutions have prompted the use of computers for knowledge discovery (Parsaye et al., 1989), (Frawley et al., 1991). Massively parallel computers, and even supercomputers, have

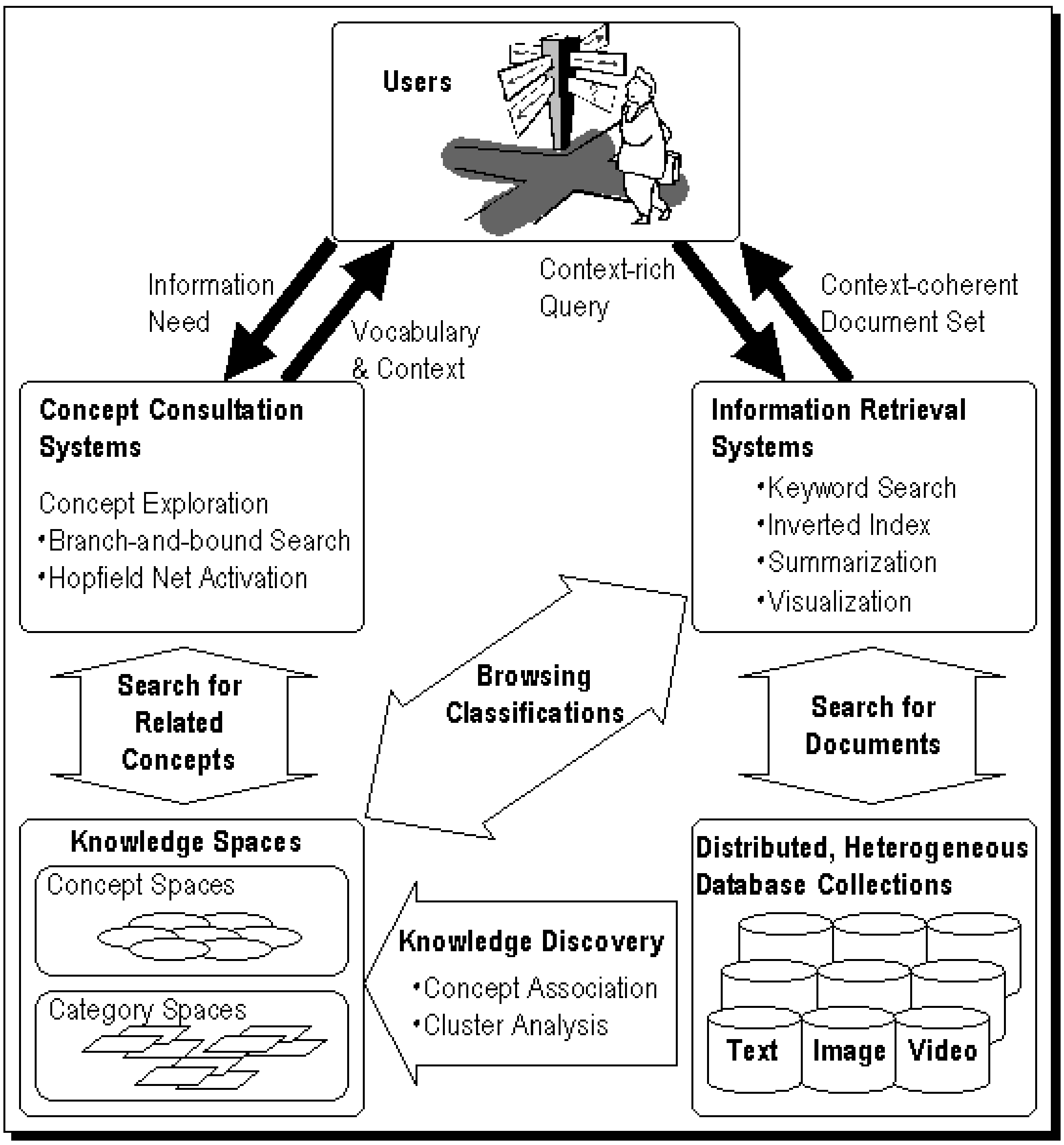

Figure 3.1: A framework for semantic-enabled information technologies

also been considered for analysis of some really large business or scientific databases.

- *Knowledge discovery methods:*

  Various knowledge discovery techniques have been developed over the past few decades by statistics, information science, and artificial intelligence researchers.

  Statistical algorithms typically examine quantitative data for the following purposes (Parsaye et al., 1989): clustering descriptors with common characteristics, e.g., factor analysis, principal components analysis (Morrison, 1976), and cluster analysis (Everitt, 1980); hypothesis testing for differences among different populations, e.g., t-test and analysis of variance (ANOVA) (Montgomery, 1976); trend analysis, e.g., time series analysis (Nelson, 1973), (Morrison, 1976); and correlation between variables, e.g., correlation coefficient and linear/multiple regression analysis (Montgomery, 1976).

  Recently, classical symbolic AI learning algorithms such as ID3 (Quinlan, 1983) and AQ (Michalski and Larson, 1978) and resurgent neural net learning algorithms such as Backpropagation (Rumelhart et al., 1986) have provided new perspectives for knowledge discovery. These techniques allow effective analysis of both qualitative and quantitative data. Unlike the statistical approach, which typically is based on some underlying models, assumptions, and stringent conditions, many AI-based techniques are more flexible, easier to use, more powerful, and produce output that is more meaningful to users. (For a complete overview of the AI-based learning techniques, readers are referred to (Carbonell et al., 1983), (Dietterich and Michalski, 1983), (Knight, 1990), (Frawley et al., 1991).)

My research focuses on a knowledge discovery method for semantic-enabled technology based on the statistical co-occurrence analysis (Chen and Lynch, 1992) or context analysis (Attar and Fraenkel, 1977) in the field of information retrieval.

- *Discovered knowledge and other knowledge sources:*

  In addition to the mathematical formulas and parameters produced by statistical techniques, symbolic AI-based techniques produce outputs that are based on traditional knowledge representation schemes such as semantic net, frame, decision trees, and logic (Parsaye et al., 1990), (Quinlan, 1983), (Michalski and Stepp, 1983). Because most AI-based knowledge representations are grounded on cognitive research (human memory, problem solving, story understanding, production systems, etc.) (Anderson, 1985a), they are often considered more natural and understandable (for users) than statistical formulas or neural nets.

The amount of knowledge discovered by various knowledge discovery methods (e.g., number of rules produced, number of nodes and links of a system-generated semantic net or neural net, levels and branches of a decision tree, etc.) sometimes may be substantial. For some applications, it may also be necessary to include such domain-specific knowledge bases, expert systems, corporate rules and guidelines, data dictionaries, and external thesauri in an institution's complete knowledge repository – shown in Figure 3.1 as the *Knowledge Space*.

My research focuses on the generation of *Concept Space* as a knowledge source and a part of the semantic-enabled information technologies.

- *Information retrieval systems:*

  Information retrieval systems are commonly known as *search engines* nowadays because of the popularity of several major Internet search engines like Yahoo!, Altavista and Northern Light (Baeza-Yates and Ribeiro-Neto, 1999). The ability to index hundreds of million of web pages does not translate well to the ability to present what users want. Oftentimes, users are overwhelmed by millions of potentially retrievable web pages. The common practice for dealing with this overloading problem is to give the top ten or twenty ranked web pages. Some better solutions work toward information visualization and summarization (Card et al., 1998), (Shneiderman, 1997), (Tufte, 1990), (Tufte, 1997). over retrieved data set.

- *Concept consultation systems:*

  Concept consultation systems are system-supported *concept exploration* or knowledge management tools for users. In order to manage and utilize the different knowledge bases within a knowledge space, it is necessary to develop some high-level, friendly, and efficient system-supported methods or tools for users. In Figure 3.1, while knowledge discovery methods help generate knowledge bases from databases; concept consultation systems make use of knowledge sources and provide users with *concept exploration* methods to manage and utilize these knowledge bases effectively.

  These concept exploration methods need to be highly *interactive* so that users can explore in the knowledge space freely and efficiently, articulate their conceptual models, and use whatever knowledge is relevant to their applications or tasks. While *knowledge discovery* methods are *data-driven*, *concept exploration* needs to be *user-driven*.

  Kaufman et al. (Kaufman et al., 1991) suggested a *knowledge management* component for knowledge discovery applications that can assist in the operation and use of knowledge bases and is similar to the *database management* function of commercial database management systems (DBMS). They proposed counterparts of the relational DBMS operators such as SELECT, PROJECT, JOIN, and INTERSECT for knowledge management. However, the proposed functionalities and roles for these operators remained vague and these authors have suggested further research investigating different high-level operators for different representation schemes. My research has been aimed at examining in detail an "ACTIVATE" operator on semantic net and neural net representations that can assist in spreading activation based inferencing. It focuses on the investigation of concept exploration methods as semantic-enabled information technology for users to communicate their information needs.

3.3.2 Structure of Dissertation

Based on the framework shown in Figure 3.1, the next three chapters describe three distinctive but closely interrelated pieces of research to investigate the creation and use of semantic-enabled information technologies to help users seek information.

Chapter four describes the use of the semantic component (concept space) to help a user express an information need. This chapter directly relates to the *concept consultation systems* in the framework. Two algorithms - branch-and-bound and Hopfield Net - are employed to explore within the concept space. The purpose of using these two algorithms is to automatically gather closely related concepts for presenting to a user who may choose to use any number of related concepts to better express a query. Experiments are also performed to evaluate the performance of each exploration methods and its use.

Chapter five describes the process of identifying and extracting concepts from textual documents as basis for further semantic analysis. A concept is represented by term phrases appearing naturally in sentences. Co-occurrence analysis, one of the context analysis techniques, is then performed on the entire document collection. This context analysis calculates the strength of relationship between each possible pair of co-occurring concepts. Essentially, it automatically builds a thesaurus of concepts, called a concept space, with statistical relationships rather than traditional semantic relationships. Experiments are performed to evaluate the performance of the algorithmical strength of relationship between concepts. This chapter corresponds with the *knowledge discovery* methods and *knowledge sources* components in the framework.

Finally, chapter six presents my conclusions and thoughts for future exploration of semantic research. Appendix A presents a detailed benchmark testing of the two algorithms used for concept exploration. Appendix B presents a sample interactive session of the concept exploration process. Appendix C shows a web-based information retrieval system called *Cancer Space*. Appendix D lists the funding sources and specific personal acknowledgements for various stages of research in this dissertation.

CHAPTER 4

CONCEPT SPACE CONSULTATION

4.1 Objectives

Knowledge discovery in databases that is based on statistical analysis, machine learning, and neural net computing has attracted attention from researchers from several disciplines. In recent years, specialized software, powerful workstations and even massively-parallel computers have been used to perform extensive knowledge discovery on real-life databases. Advancement in hardware technology and the continuous development of practical, multiple-discipline, "intelligent" analysis techniques has made "knowledge discovery" a highly promising area for information systems research and practice in the next decade.

With the computation power of prevailing hardware and the "intelligence" of many practical algorithms, knowledge discovery has also made possible the development of large knowledge bases. Knowledge discovery algorithms can explore and identify the underlying patterns in large databases and create much larger knowledge bases than it is possible to develop using manual, labor-intensive knowledge elicitation (Parsaye et al., 1989). The resulting *discovered knowledge* can also be consolidated and used in conjunction with other existing knowledge sources (either manually created or extracted from other sources). The surge in knowledge base development and rapid increases in the size of knowledge bases have prompted researchers to suggest *knowledge management systems* as a counterpart to *database management systems* (Kaufman et al., 1991). The amount and diversity of discovered knowledge have called for the development of high-level, efficient knowledge management tools.

A *knowledge network* may consist of knowledge discovered from real-life databases and knowledge extracted from existing domain-specific knowledge sources. This chapter presents research concerning algorithmic *concept exploration* in a large network of knowledge. I propose two spreading activation based algorithms for *concept exploration*. One is based on a conventional, serial branch-and-bound search algorithm and the other on neural net *parallel relaxation*. These algorithms can traverse (explore) a knowledge network automatically and suggest to users a set of concepts most relevant to their applications. I believe this automatic *concept exploration* component can help alleviate the cognitive demand often associated with the manual browsing process and make the large-scale output of *knowledge discovery* methods more accessible and useful. Effectively, this *concept exploration* component provides a layer of technology acting as a mediator between users and semantics-bearing information inside various knowledge bases.

4.2 Research Questions and Methodology

The specific research questions to be investigated were:

- **Question 1**: Would the *automatic concept exploration process* be able to help users identify more relevant concepts?
- **Question 2**: Would such a process be able to perform more efficient exploration of a concept space than the conventional manual browsing method?
- **Question 3**: If so, which algorithmic methods - symbolic-based branch-and-bound or neural network-based Hopfield net algorithm - is better in terms of gathering relevant concepts from knowledge sources?
- **Question 4**: Would the *concept space consultation* process provide a *semantic medium* to reduce the *cognitive demand* from users in terms of *elaborating information needs*?
- **Question 5**: Would the *concept exploration process* be able to help users find more relevant documents?

I used the systems development methodology to develop a prototype system which accessed various knowledge sources using manual browsing, a branch-and-bound searching algorithm, or a Hopfield net algorithm. I also used the experimental design methodology to collect quantitative measures like term/document recalls and precisions. Qualitative information was collected through interviews after the subjects had completed the last two phases of the experiment.

4.3 **Background and Issues**

4.3.1 Spreading Activation in a Concept Network

In this research, the focus was on concept exploration methods in a large knowledge *network* structure. Two spreading activation based search methods were developed, one built on symbolic, serial branch-and-bound search and the other on the neural net parallel relaxation method, which can be used, respectively, in a semantic net or a neural net knowledge representation. The research represents a step toward developing a practical and useful *knowledge management system* (KMS).

Among the knowledge representation schemes frequently adopted in knowledge discovery research, *network* based representation often has been considered one of the most direct and natural representations. Within a network of inter-connected nodes and directed links, the relationships between objects, knowledge, and patterns of interest can be represented explicitly. Two main paradigms of network representation needed to be examined in detail in the context of our research: the symbolic AI-based semantic nets (SN) and the connectionist AI-based neural nets

(NN). While both representations continue to share some common features and exhibit some historical peculiarities, differences between them have become blurred as network topologies and hybrid systems have proliferated.

4.3.2 Semantic Nets

A semantic net is a structure for representing knowledge as a pattern of interconnected nodes and links (Sowa, 1991a). Modern semantic nets are often considered outgrowths of Quillian's work on "Semantic Memory" (Quillian, 1968). Since then, many different versions have been implemented. Although the terminology and notations may vary, the following characteristics are common to most of them (Sowa, 1991a):

- Nodes represent concepts of entities, attributes, events, and states.
- Links represent conceptual relationships that hold between concept nodes. Labels on the links specify relationship types.

Over the past two decades, many semantic net researchers have attempted to develop a formal theory and models of semantic nets. For example, it is possible to translate a semantic net into its equivalent first-order logic representation (Charniak, 1981). It has been shown that the class of semantic net languages cannot be be differentiated from the class of non-semantic net languages on the basis of representational adequacy (Shastri, 1991). More recently, some semantic net researchers have stressed the importance of developing efficient inferencing algorithms for real tasks on semantic nets. Bill Woods, one of the pioneers in semantic net research remarked:

> A major gap, I believe, is the lack of sufficient emphasis on algorithmic uses of network representations to support various kinds of inference (Sowa, 1991b).

Many other researchers have also suggested developing efficient, real-time algorithms for inferencing, adopting semantic distance between two concepts as spreading activation heuristics, and measuring the performance of inferencing algorithms against real-life, large-scale applications (Sowa, 1991b), (Shastri, 1991).

The basis of most inferencing methods on semantic nets, however, is *spreading activation*, which is considered a variant of the *state space traversal* adopted in most symbolic AI-based systems (Winston, 1984), (Rich and Knight, 1991). Inference is performed by traversing (activating) the links and nodes connected to some initial nodes of concepts, with shorter paths considered preferable to longer ones, a characteristic of human reasoning (Anderson, 1985a), (Shastri, 1991). Conventional *search* techniques including depth-first-search (DFS), breadth-first-search (BFS), branch-and-bound search, and A^* search have often been used for state space traversal in applications such as the traveling salesman problem (Winston, 1984).

In this research, a branch-and-bound spreading activation algorithm was adopted for the following reasons. A branch-and-bound algorithm is considered an "optimal" search method, which aims at obtaining the shortest possible path during search. Identifying the shortest path in a large knowledge network is important because the objective of search is to find other very relevant concepts (i.e., neighboring concepts), instead of just any other concepts in the network. The A^* and branch-and-bound were selected initially instead of the more popular (but non-optimal) DFS or BFS method, but A^* search was later abandoned because it lacks a proper *under-estimate* measure in the concept exploration domain (A^* terminates faster than branch-and-bound due to its under-estimate measure).

Branch-and-bound's systematic exploration of the neighboring structure of some initial nodes based on a priority queue made it suitable for the concept exploration application. Cohen and Kjeldsen's "constrained spreading activation" (Cohen and Kjeldsen, 1987) and Chen and Dhar's METACAT (Chen and Dhar, 1991) all incorporated branch-and-bound's serial, optimal search property to some degree.

4.3.3 Neural Nets

Neural nets which represent knowledge, objects, and patterns in terms of interconnected nodes and weighted links have made an impressive come-back in recent years. Among several reasons for this are the appearance of faster computers that can simulate larger networks and the development of new neural net architectures for real-life applications.

Neural nets are similar to semantic nets in terms of network representation, but *distributed* representations of neural nets (Hinton and Sejnowski, 1986) do not use individual nodes to represent concepts or links to represent conceptual relationships. Rather, they use patterns of activations over many units in the network. For example, a Hopfield net (Hopfield, 1982) may provide a distributed representation for a content-addressable memory in which each structure is stored as a collection of active units. Such special *distributed* representation allows the network to be more damage-resistant, a property that exists in animal memory. In addition to possessing this distributed representation property, neural nets perform *parallel relaxation* search, during which nodes are activated in parallel and are traversed until the network reaches a stable state. This process often is considered more efficient than serial, symbolic search because it makes use of states that have no analogues in symbolic search and because it maps naturally onto highly parallel hardware (Rich and Knight, 1991).

The Hopfield net's parallel search on a single-layered network of nodes and weighted links and its convergence property made it suitable for automatic concept exploration. With the activation of some initial nodes, the Hopfield algorithm can activate their direct neighbors at the next iteration, combine and compute activation values from various sources, and continue the iterations until the activation strengths "die" out (a gradual damping process). In essence, the Hopfield net identifies other relevant concepts through a parallel and convergent approach that recently has been used for information retrieval applications (Chen et al., 1993); its novel search capability makes it ideal for concept exploration. Other popular neural networks such as Backpropagation networks or Kohonen networks (Lippmann, 1987) were not considered for this research either because of inadequate network topology (i.e., both networks consist of multiple layers of objects) or of inappropriate network activation algorithms (e.g., the Delta rule in Backpropagation is more suitable for learning). More details about the branch-and-bound and the Hopfield implementations adopted will be presented in Section 4.6.

Many neural net and semantic net researchers consider *localist* representations of neural nets to be a variant of semantic nets (or conversely, semantic nets to be a variant of neural nets), in which each node and link respectively represents an individual concept and a conceptual relationship (Sowa, 1991b), (Bechtel and Abrahamsen, 1991), (Rich and Knight, 1991).

Systems developed by AI researchers frequently demonstrate the similarities of localist neural net and semantic net knowledge representations. For example, Anderson's ACT* nets (Anderson, 1983) and Fahlman's NETL (Fahlman, 1979) both use nodes for concepts and allow some algorithmic spreading activation on the networks. Many hybrid systems developed in recent years employ symbolic and neural net characteristics. For example, Touretzky and Hinton (Touretzky and Hinton, 1988) and Gallant (Gallant, 1988) proposed connectionist production systems, and Derthick (Derthick, 1988) and Shastri (Shastri, 1991) developed different connectionist semantic networks.

This research investigated a localist representation of a neural net which can also be perceived as a semantic net. To avoid confusion, a more generic term, *knowledge network*, is used in the remainder of the paper. The research examined the effects of implementing two different spreading activation methods: symbolic branch-and-bound versus connectionist parallel relaxation on a large hybrid neural-semantic-net. Implementation issues such as computational efficiency, scalability of the algorithms for use in large-scale networks, and the performances of the two algorithms will be presented in detail.

4.4 Concept Exploration in a Large Text-based Concept Network

The proposed framework in my dissertation plan was applied to a research environment where *knowledge discovery* and *concept exploration* were essential for information retrieval and intelligence analysis. The entities involved in this research environment consisted of many international computing researchers (users), a couple of document databases, and several knowledge sources. The findings regarding automatic construction of networks of concepts for this application has previously been reported in (Chen and Lynch, 1992). A blackboard design for integrating heterogeneous knowledge bases has also been reported in (Chen et al., 1993), who presents an overview of the application, emphasizing the *concept exploration* component in this environment.

4.4.1 Databases

The organization studied is the Mosaic research group at the University of Arizona, whose members have conducted research over the past decade in the areas of foreign-nation studies and assessment of information technologies, focusing on the (former) Soviet Union and Eastern Europe (Russian/EE) (Goodman et al., 1990), (McHenry et al., 1990). Group members (analysts) collect articles and other forms of international computing-related academic publications, browse and study (foreign) documents collected, exchange ideas with foreign researchers via e-mail, telephone, and other means, periodically visit foreign countries and organizations, and attend major international conferences and professional meetings. They build their knowledge around certain subject areas, develop their own personal contacts with foreign researchers and organizations, and shape their beliefs, values, and judgments concerning international computing technologies and developments in specific countries of interest.

A significant portion of the Mosaic group memory and expertise has been captured by the Mosaic document databases. A custom-made information storage and retrieval system, built on top of INGRES, supports Mosaic research (Lynch et al., 1990). The two databases considered most important to the Mosaic research environment both reside in the INGRES database management system. The "Russian" database, created manually, contains about 40,000 documents (article abstracts, newspaper articles, electronic mail exchanges, business cards, etc.) in a database of about 200 megabytes. In addition, Mosaic analysts also have extracted abstracts of recent computing-related articles from the DIALOG database, which they call the Public database. It consists of about 3,000 articles (20 megabytes). Some indexes had already been assigned to these documents by the DIALOG database. Because of operational concerns, this research only experimented with the Public database.

4.4.2 Knowledge Discovery Methods

A previous paper (Chen and Lynch, 1992) reported detailed findings about *automatic generation of knowledge bases* from document databases. Several new cluster analysis algorithms were developed to produce knowledge bases (Salton, 1988), (Everitt, 1980), (Chen et al., 1993). These algorithms were based on the frequency of terms co-occurring in the documents and the resulting knowledge was captured in a semantic net representation where nodes represent different types of concepts and weighted links indicate their strengths of relevance.

In this thesis, I use terms and indexes (terms used for indexing) interchangeably. When describing indexes and terms in the context of the semantic net or neural network representation, I also refer to them as nodes or concepts (to be consistent with the artificial intelligence terminology). The procedure used for automatically creating knowledge bases is sketched below. (Readers are referred to (Chen and Lynch, 1992) for details.)

1. *Determine unique indexes:* All indexes assigned to all the documents are identified in the database (assigned previously by human indexers or generated automatically by automatic indexing techniques (Salton, 1988), (Chen and Lynch, 1992)).

2. *Weight computation:* For each unique index, its term co-occurrence probabilities with all other indexes are computed based on the asymmetric "Cluster Function" developed by the authors (Chen and Lynch, 1992). The term co-occurrence probability, which is a real number between 0 and 1, indicates the "relevance" weight between any two indexes.

   $Weight(T_j, T_k) = \frac{\sum_{i=1}^{n} d_{ijk}}{\sum_{i=1}^{n} d_{ij}}$

   $Weight(T_k, T_j) = \frac{\sum_{i=1}^{n} d_{ijk}}{\sum_{i=1}^{n} d_{ik}}$

   They indicate the similarity weights from T_j to T_k (the first equation) and from T_k to T_j (the second equation). Where d_{ij} indicates index T_j in document i (value: 0 or 1), d_{ik} indicates index T_k in document i (value: 0 or 1), and d_{ijk} indicates both indexes T_j and T_k are in document i (value: 0 or 1).

The limitation of the popular symmetric co-occurrence coefficients, e.g., cosine, Dice, and Jaccard's, have been reported by Peat and Willett (Peat and Willett, 1991). Their research showed that similar terms identified by symmetric co-occurrence functions tended to occur very frequently in the database that was being searched and thus did little or nothing to improve the discriminatory power of the original query. They concluded that this can help explain Sparck Jones's finding that the best retrieval results were obtained if only less frequently occurring terms were clustered and if more frequently occurring terms were left unclustered (Sparck Jones, 1971). Research developing several thesauri for capturing subject experts' domain concepts (in terms of keywords and relationships) for several applications reached the same conclusion (Chen and Lynch, 1992), (Chen et al., 1995). Especially in the Chen and Lynch's research (Chen and Lynch, 1992), the asymmetric function out-performed the cosine function in a generating domain-specific thesaurus.

4.4.3 Knowledge Sources for Concept Network

The Public knowledge base (the Public KB) was used for testing the *concept exploration* algorithms. The size and the subject area (general computing) of the Public knowledge base provided the ease to implement and evaluate the algorithms on the hardware platform. The Public knowledge base contains 1,488 concepts (nodes) and 44,496 weighted relationships (links). The weights associated with the links indicate the "strength" of relevance between two terms in the network. These weights are probabilities between 0 and 1.

Use of a thesaurus or a knowledge base for "intelligent" information retrieval and management has been the focus of research in which many information science and computer science researchers have attempted to capture experts' domain knowledge for information retrieval or information management. For example, CoalSORT (Monarch and Carbonell, 1987), a knowledge-based interface, facilitates the use of bibliographic databases in coal technology. A semantic net, representing an expert's domain knowledge, embodies the system's intelligence. GRANT, developed by Cohen and Kjeldsen, is an expert system for finding sources of funding for given research proposals (Cohen and Kjeldsen, 1987). Its search method - constrained spreading activation in a semantic net - makes inferences about the goals of the user and thus finds information not explicitly requested but likely to be useful. Chen and Dhar incorporated a portion of the *Library of Congress Subject Headings* into the design of an intelligent retrieval system (Chen and Dhar, 1991). The system adopted a heuristics-based spreading activation algorithm to assist users in articulating their queries. The National Library of Medicine's Unified Medical Language System (UMLS) may be the largest-scale effort in integrating different knowledge sources (Humphreys and Lindberg, 1989), (McCray and

Hole, 1990), (Lindberg and Humphreys, 1990). It includes a Metathesaurus, a Semantic Network, and an Information Sources Map. The Metathesaurus contains information about biomedical concepts and their representation in more than 10 different vocabularies and thesauri. The Semantic Network contains information about the types of terms (e.g., "disease," "virus," etc.) in the Metathesaurus and the permissible relationships among these types. The Information Sources Map contains information about the scope, location, vocabulary, and access conditions of biomedical databases of all kinds.

In an attempt to expand the knowledge coverage of our knowledge network, two more external knowledge sources (in the forms of thesauri) were included in the implementation: the complete ACM Computing Review Classification System and a portion of the Library of Congress Subject Headings. The ACM Computing Review Classification System (ACM CRCS) represents the general computing categories used by the ACM for classifying the computing literature. The Library of Congress Subject Headings (LCSH) represents general computing terms selected by the Library of Congress for classifying computing-related books. Each knowledge source has its unique structure and vocabulary.

The ACM CRCS is based on a hierarchical structure. Four levels of specificity exist. Terms fan out level by level. Although its classification structure is simpler and its subjects are less specific, it does represent general computing terms and their relationships very nicely. Two types of terms from the ACM CRCS were identified. The first type deals with specific topics which are similar to the Library of Congress subject headings, e.g., “information retrieval systems” and “machine learning.” The second type of term, however, indicates general computing-related categories. These categories can be appended to terms in the ACM thesaurus. Examples of general categories are: verification, documentation, testing, etc. 18 general categories and 1,141 specific terms are identified from the ACM thesaurus. Five types of relationships were identified: BT/NT (broader/narrower term) indicates hierarchical relationships between the specific terms, RT (related term) indicates an associative relationship (this relationship is shown in the parentheses following some terms), and ISA/INST (is-a and instance-of relationships) indicates the relationships between specific terms and general categories. For example, “Microprogram Design Aids – Verification” (a specific term) is-a kind of “Verification” (a general category) or conversely, an instance-of “Verification” is “Microprogram Design Aids – Verification.” ISA and INST can be considered as special cases of the BT and NT relationships, respectively. A total of 2,922 relationships was captured from the ACM Computing Review Classification System.

The LCSH is network based and contains terms and cross-references between terms. Terms indicate topics. Five types of relationships exist between terms: USE/UF (use or used for) indicates a synonymous relationship, RT (related term)

				Public		ACM CRCS		LCSH	
Terms in	nodes	links	links/node	also-in	%-in	also-in	%-in	also-in	%-in
Public	1,488	44,486	29.9	-	-	122	8.2%	177	11.9%
ACM CRCS	1,157	2,918	2.5	122	10.5%	-	-	116	10.0%
LCSH	10,972	32,702	2.9	177	1.6%	116	1.1%	-	-

Table 4.1: Terms/links in knowledge bases

indicates an associative relationship, and BT/NT (broader/narrower term) indicates hierarchical relationships. Our subset of the Library of Congress Subject Headings contains 10,972 terms and 32,702 relationships.

A summary of the structures of the three knowledge sources in terms of the frame-based representation is presented in Figure 4.1. Collectively, they augment the *knowledge* of our system and they can be very useful in assisting searchers in articulating their queries and improving search recall. An analysis of the terms appearing in the three knowledge bases revealed that only a few hundred terms appeared in two or more knowledge sources (see Table 4.1). For example, Row 3, Columns 7 and 8 show that 122 terms in the Public KB also appeared in the ACM CRCS, constituting 8.2% of the ACM CRCS terms. The complete knowledge network can be perceived as the union of three partially-overlapping networks – the Public knowledge base, the ACM CRCS, and the LCSH.

```
Public KB Object Frame:
  {Object:
      Object type: (term)
      RT: (list of related terms)
  }

ACM CRCS Object Frame:
  {Object:
      Object type: (term)
      NT: (list of narrower terms)
      BT: (list of broader terms)
      RT: (list of related terms)
      ISA: (list of parent terms)
      INST: (list of children terms)
  }

LCSH Object Frame:
  {Object:
      Object type: (term)
      NT: (list of narrower terms)
      BT: (list of broader terms)
      RT: (list of related terms)
      USE/UF: (list of synonymous terms)
  }
```

Figure 4.1: Frame-based representations for the knowledge sources

4.4.4 Concept Exploration Methods

The rich semantics and cross-references provided in various knowledge bases enable users of such systems to get into a network of knowledge easily and to explore and navigate in this network. However, to perform query refinement and to identify relevant concepts efficiently and effortlessly in a large network/hierarchy of concepts (perhaps several thousand to a few million) and at the same time avoid both the classical hypertext "embedded digression problem" (a system can potentially confuse and disorient its user) and the "art museum phenomenon" (a system can cause users to spend a great of time while learning nothing specific) (Foss, 1989), (Carmel et al., 1992), requires an active and intelligent way to traverse multiple thesauri and multiple links. In the National Library of Medicine's Unified Medical Language Systems (Humphreys and Lindberg, 1989), (McCray and Hole, 1990), (Lindberg and Humphreys, 1990), which contains several million biomedical concepts and their relationships from more than 10 sources, browsing could become extremely cognitively demanding for users. System-aided concept exploration and multiple thesauri consultation have become a pressing research issue.

"Spreading activation," a memory association mechanism that originated in human memory research, has been used successfully in various semantic net and neural net applications. Our application, which includes networks based on labelled links (the LCSH and the ACM CRCS) and weighted links (the Public KB), is considered a hybrid system of semantic nets and neural nets.

In order to allow seamless spreading activation across all three knowledge bases, a weight propagation scheme was developed to assign normalized weights to the labelled links (e.g., NT, RT, BT, etc.) based on the weights associated with the Public KB. This scheme enables the traversal of the resulting loosely-coupled knowledge network by means of either symbolic state space search or neural net parallel relaxation. The following outlines the weight assignment and propagation scheme:

1. *Elicit activation criteria:* Two activation criteria need to be supplied by the user: weights assigned to individual knowledge sources and weights assigned to different types of links. A scale of 0 to 10 for each knowledge source is used to indicate the searcher's preferred sources – 0 indicates the lowest preference level (i.e., the source is considered irrelevant) and 10 indicates the highest preference level. The ratings provided are used to determine the relative weights associated with different knowledge sources. Another 0 to 10 scale is also used to elicit the user's preferences among three types of links: broader term (BT), narrower term (NT), and related terms (RT). For example, assigning a higher rating to NT than to BT indicates the user's intention to traverse toward more specific concepts (through NT links). These ratings can be used by the spreading activation algorithms to determine the activation direction. By allowing users to indicate their preferred knowledge sources and link types, the activation algorithms can traverse the knowledge sources more efficiently and "intelligently." Setting these weights is a straightforward

task for most users – the system prompts them to enter a numeric value for each parameter. The 0-10 scale allows them to indicate the relative importance of their preferences. Default values can also be used if a user chooses not to change the setting. (In performing benchmark testing, discussed in Appendix A, and the user evaluation experiment, discussed in Section 4.7, default values for these weights were set as follows: Public/ACM/LCSH = 10/10/10 and RT/NT/BT = 3/10/1.)

2. *Propagate connection weights:* For the Public KB, link weights have already been generated and stored. Because all three knowledge sources have some form of related term (RT) relationship, the RT link weights in the Public KB can be used as the basis for assigning weights to the RT links in the other two knowledge sources. The knowledge source and link type activation criteria obtained from the prior step can then be used to modify the assigned weights to reflect the user's search criteria. For example, the NT link weights were computed as the product of the average RT weight in the Public KB and the relative NT/RT weight solicited from users (i.e., ART * LW(NT)/LW(RT), as shown in the equations below). The same process was applied to the BT links. A sketch of the connection weight assignment for the different knowledge sources is shown in Figure 4.2 and the detailed spreading activation algorithms on this knowledge network are discussed in the next section.

```
let knowledge sources weights be:
  KW(Public) : KW(ACM) : KW(LCSH) = a : b : c;
                                            {Solicited from users.}
let link weights be:
  LW(RT) : LW(NT) : LW(BT) = x : y : z;
                                            {Solicited from users.}
let ART :=  the average weight of the RT links in the Public KB;
                                            {Computed from INGRES relations.}
Propagate weights to links in ACM CRCS:     {b/a: relative weight for ACM.}
  LW(RT) := b/a * ART;
  LW(NT) := LW(INST) := b/a * (ART * y/x);  {INST is special case of NT.}
  LW(BT) := LW(ISA)  := b/a * (ART * z/x);  {ISA is special case of BT.}

Propagate weights to links in LCSH:         {c/a: relative weight for LCSH.}
  LW(RT) := c/a * ART;
  LW(NT) := c/a * (ART * y/x);
  LW(BT) := c/a * (ART * z/x);
  LW(USE/UF) := 1;                          {Weight for synonymous link is 1.}
```

Figure 4.2: Connection weight assignment for the knowledge sources

In conclusion, with multiple, heterogeneous thesauri, created either manually or automatically for experimental purposes, I was able to examine the feasibility of adopting spreading activation algorithms for "concept space consultation." However, the proposed framework and algorithms were intended for the more general network-based knowledge inferencing tasks.

4.5 Two Algorithms for Spreading Activation

In this section, the two algorithms for automatic and "intelligent" concept exploration in the knowledge network presented above are discussed in detail. The novel features of the algorithms and a comparison of their behaviors also are described.

4.5.1 A Branch-and-bound Spreading Activation Algorithm: Semantic Net Based

Branch-and-bound search has been used frequently in state space traversal for identifying *optimal* paths (Winston, 1984). Applications such as scheduling, network routing, and the traveling salesman problem typically adopt this search method. As explained earlier, branch-and-bound was chosen over the more popular and simple DFS or BFS method because of its "optimal" search property, which was essential for finding the shortest paths and identifying a set of most relevant concepts in the knowledge network. Chen first reported the use of such a method for information retrieval (Chen and Dhar, 1991). The algorithm automatically traversed an online thesaurus (the Library of Congress Subject Headings) and made term suggestions.

Our branch-and-bound implementation starts with terms provided by the user. These starting terms are assigned a value of 1 as their node weights by the algorithm. (The prototype system requires an exact match between a search term and a node in the knowledge network. However, this can be improved in future development.) The terms are then used to activate their directly linked neighbors. Each activated neighbor receives a weight equal to the product of the weight of the activating node and the link weight. Based on the basic data structure adopted in branch-and-bound search, all activated nodes are put into a priority queue according to their associated weights. Terms with the heaviest weight in the queue are then used to activate their neighbors – terms which have equal weights are activated at the same time. Each node also records its starting term. Each activation is considered an *iteration*.

Branch-and-bound spreading activation repeats until a desirable user-defined state is reached (the stopping condition). When interacting with the algorithm adopted in the thesaurus consultation experiment, users are requested to provide a desired number of system-suggested associated terms, say x. This user-specified number is used to determine the stopping condition for the branch-and-bound iterations. The algorithm first computes the associated node weights generated from the first iteration and then uses the desired number of terms to determine a cut-off threshold. The cut-off helps obtain x terms which are greater than the threshold. During the next iterations, the system performs the branch-and-bound routine and combines weights of paths which originated from different starting terms. Nodes which obtain higher weights will thus be placed at the front of the priority queue.

In essence, the algorithm uses a user-specified number of terms to determine a stopping threshold for branch-and-bound. During iteration, it activates the highest-ranked nodes, computes node weights based on a simple multiplication function, and combines weights if the node can be reached from different starting terms. A more detailed sketch of the branch-and-bound spreading activation algorithm follows:

1. **Assigning weights to symbolic links:**

   The initial status of the net is represented by the weighted links and nodes associated with the three thesauri as discussed earlier. t_{ij} represents the weight from node i to node j.

2. **Initialization with user's input:**

   An initial set of starting terms $\{S_1, S_2, ..S_m\}$ is chosen by the user. Each node in the network (of n nodes) which matches the starting terms is initialized to have a weight of 1:

$\mu_i(0) = x_i,\ 0 \le i \le n-1$

$\mu_i(t)$ is the weight of node i at iteration t. At time 0, the nodes corresponding to the starting terms are assigned 1.

The algorithm then creates a priority queue, $Q_{priority}$, based on the decreasing weights assigned to each node. Initially,

$Q_{priority} = \{S_1, S_2, ...S_m\}$

It also creates an output queue, Q_{output}, to store the activated nodes during each iteration. Initially,

$Q_{output} = \{\}$

3. **Activation, weight computation, and iteration:**

During each iteration, the algorithm removes the highest-weighted nodes in $Q_{priority}$, activates their neighboring nodes, and computes their neighbors' weights as follows:

$\mu_j(t+1) = \mu_i(t) \times t_{ij}$

As explained earlier, weight assignment is based on the product of the activating node weight and the link weight between the activating node and its neighbor. Recently activated nodes which had not been recorded earlier in Q_{output} are inserted into the output queue, Q_{output} (according to the order of their arrival). After computation, all expanded nodes in $Q_{priority}$ are re-sorted. In order to reward a node which can be reached by different starting terms (i.e., two different paths lead to the same node), the algorithm sums up its associated weights and assigns the result to the node. The algorithm then records this higher-weighted node in the priority queue. This heuristic of assigning higher weight to a node which can be reached from different starting nodes in the network has also been adopted in other spreading activation based systems (Shoval, 1985), (Cohen and Kjeldsen, 1987), (Chen and Dhar, 1991).

4. **Determining stopping condition:**

   The algorithm solicits an expected number of system-suggested terms (p) from the users. This number is used by the algorithm to determine the stopping condition for the branch-and-bound search. After the first iteration – all starting terms are activated because they have the same weights – $Q_{priority}$ records the direct neighbors of all starting terms in decreasing weights. The algorithm identifies the pth node in $Q_{priority}$ and obtains its weight, w_p, as the threshold for stopping the branch-and-bound activation process (one of the stopping conditions). For most queries, p terms are produced after the first iteration. However, occasionally multiple iterations are needed to obtain

p terms in $Q_{priority}$. If the system is unable to produce p terms after complete iteration (i.e., no more neighbors), the algorithm terminates.

This user-specified was established to help generate the top p terms relevant to the users' queries. During iteration, some terms which have higher weights than the terms in the queue will take up their positions in $Q_{priority}$. The algorithm stops when the output queue, Q_{output}, consists of more than p nodes (there may not be exactly p nodes in the queue because the algorithm activates all highest-weighted nodes at the same iteration) or when the highest weight in $Q_{priority}$ is less than the user-defined threshold value, w_p, or when $Q_{priority}$ becomes null (i.e., all neighbors are exhausted).

The above branch-and-bound spreading activation algorithm is in essence a serial, optimal state space search process, during which "best" nodes get activated first. The user-supplied stopping condition allows the system to decide its exploration effort based on the users' expectation. These features compare favorably with those of other heuristics-based spreading activation systems (Cohen and Kjeldsen, 1987), (Shoval, 1985), which typically neither have a user-supplied threshold nor exhibit the branch-and-bound optimal search characteristics.

4.5.2 A Hopfield Net Spreading Activation Algorithm: Neural Net Based

The Hopfield net (Hopfield, 1982), (Tank and Hopfield, 1987), a classical method of inferencing in a single-layered, weighted network, presents an interesting and novel alternative to the serial state space traversal of the symbolic branch-and-bound algorithm. It performs a *parallel relaxation* search, during which nodes are activated in parallel and activation values from different sources are combined for each individual node. Neighboring nodes are traversed in order until the activation levels of nodes on the network gradually die out and the network reaches a stable state (convergence). As discussed earlier, a Hopfield net was chosen over other neural networks because of its parallel search and convergent properties and its single-layer topology (most other neural networks contain multiple layers of objects). A Hopfield net had been used successfully for various classification and optimization tasks (Lippmann, 1987), (Simpson, 1990) and was also adopted in a blackboard-based retrieval system (Chen et al., 1993). Its search behavior in large knowledge network, however, has not been examined in detail, especially in comparison with the more traditional serial search method.

A Hopfield net can be used as associated memory, where unknown input patterns (e.g., fuzzy queries) can be classified and disambiguated based on the knowledge embedded in the network. The weighted network of knowledge sources can be perceived as inter-connections of neurons and synapses in the Hopfield net, where neurons represent concepts and synapses represent weighted links between pairs of concepts. The implementation incorporates the basic Hopfield net iteration and convergence ideas. However, significant modification also have been made to take into consideration the three different knowledge sources and other unique characteristics of our application.

Once the initial inputs and the weights have been associated with the nodes in the knowledge sources, the algorithm activates neighboring terms, combines weighted links, performs a transformation function (a SIGMOID function, f_s), and determines the outputs of newly activated nodes. The process repeats until node outputs remain unchanged with further iterations. The node outputs then represent the concepts that best describe the initial search terms. A sketch of the Hopfield net activation algorithm follows:

1. **Assigning synaptic weights:**

   The "training" phase of the Hopfield net is completed when weights have been propagated to all knowledge bases as discussed earlier. t_{ij} represents the "synaptic" weight from node i to node j.

2. **Initialization with user's input:**

   An initial set of starting terms $\{S_1, S_2, ..S_m\}$ is chosen by a user and each node in the network that matches the starting terms is initialized to have a weight of 1. Users also need to supply a desired number of suggested terms, p, as in the branch-and-bound method.

   $\mu_i(0) = x_i$, $0 \leq i \leq n-1$

$\mu_i(t)$ is the output of node i at time t and x_i which has a value between 0 and 1, indicates the input pattern for node i. At time 0, all input nodes are assigned 1.

3. **Activation, weight computation, and iteration:**

$$\mu_j(t+1) = f_s[\sum_{i=0}^{n-1} t_{ij}\mu_i(t)],\ 0 \leq j \leq n-1$$

where f_s is the continuous SIGMOID transformation function as shown below (Knight, 1990), (Dalton and Deshmane, 1991).

$$f_s(net_j) = \frac{1}{1 + \exp[\frac{-(net_j - \theta_j)}{\theta_0}]}$$

where $net_j = \sum_{i=0}^{n-1} t_{ij}\mu_i(t)$, θ_j serves as a threshold or bias and θ_0 is used to modify the shape of the SIGMOID function.

This formula shows the *parallel relaxation* property of the Hopfield net. At each iteration, all nodes are activated at the same time. The weight computation scheme, $net_j = \sum_{i=0}^{n-1} t_{ij}\mu_i(t)$, is also a unique characteristic of the Hopfield net algorithm. Based on parallel activation, each newly activated node computes its new weight based on the summation of the products of the weights assigned to its neighbors and their synapses.

4. **Convergence stopping condition:**

The above process is repeated until there is no change in terms of output between two iterations, which is accomplished by checking:

$$\sum_{j=0}^{n-1} |\mu_j(t+1) - \mu_j(t)| \leq \epsilon$$

where ϵ is the maximal allowable error (a small number). The final output represents the set of terms relevant to the starting term.

The algorithm presents the top p terms among the final activated nodes if the number of final activated nodes is greater than p, the user's expected number of terms. If the number of final activated nodes is less than p, the system repeats the complete activation process by adopting a set of lower thresholds – lower θ_0 and θ_j values – in order to derive more activated nodes. In the implementation, the system is allowed to lower its thresholds three times, incrementally. If no more terms can be derived after lowering the thresholds three times, the algorithm terminates and presents the results from the last activation. From the implementation experiments, it appears that these default thresholds were able to handle most user requests. Only rarely did the system need to lower its thresholds more than once or twice.

This threshold-tuning effort was critical to this application as it was in other Hopfield net applications (Lippmann, 1987), (Knight, 1990). The objective in tuning was to obtain a manageable number of branches at each iteration and a reasonable number of iterations. After experimentation, a default value of 0.11 for θ_j and a value of 0.05 for θ_0 were selected as global default values for the network. The other 3 sets of (θ_j, θ_0) were: (0.065, 0.047), (0.056, 0.0464), and (0.047, 0.0458), respectively.

Both the branch-and-bound and the Hopfield net activation algorithms were developed in C and run on a DECStation 5000/120 (25 MIPS machine, ULTRIX, 1.2 gigabyte disk). The three knowledge sources were stored as three individual flat files in our testing.

4.6 System Evaluation: A Benchmark Testing

In order to examine the novel characteristics and performances of our two algorithms, I performed a benchmark testing and a user evaluation experiment, respectively. The aim of the benchmark testing was to reveal the computational characteristics of the two algorithms, specifically the number of iterations performed by each method, the computing times, and the "source of knowledge" for each system-suggested term. More details about the benchmark testing design and results appear in Appendix A.

The user evaluation detailed in next section, on the other hand, aimed at addressing performance issues. In order help readers perceive a sense of how the two spreading activations were being used in the consultation process, sample system sessions based on the two algorithms are presented in Appendix B.

4.7 User Evaluation: A Concept Exploration Experiment

The user evaluation experiment was designed to reveal the "quality" of the algorithms' suggestions. I aimed to find out whether the two algorithms were able to help identify more relevant terms and documents and perform more efficiently than the conventional hypertext-like browsing method (i.e., manual thesaurus browsing). Term recall/precision, document recall/precision, and time spent were used as performance measures. I also recorded the subjects' verbal comments on their retrieval operations. The Public database and three computing-related knowledge bases were used in the experiment.

4.7.1 Experimental Design

Three subjects were enlisted for this experiment, two of whom were advanced (3-4 year) Ph.D. students in an Information Systems Department and one was a M.S. student in a Library School. All were working on theses in database, artificial intelligence, or information retrieval related areas. Six dissertation abstracts which appeared in the 1991 Spring issue of SIGIR Forum (a publication of the

ACM special interest group on information retrieval) were selected for testing. These dissertations were all in areas that were familiar to the subjects with titles such as “Retrieval by Similarity in a Knowledge Base of Reusable Code,” “Cognitive Aspects of Human-Computer Interaction: Mental Models in Database Query Writing,” etc. This experiment was conducted in three sequential phases.

Phase 1: The subjects were shown the title and a half-page abstract for each dissertation. They were asked to read this information carefully and were requested to identify the subject areas or topics that they thought they would need to explore in order to develop a comprehensive overview and understanding of the dissertation. An experimenter recorded the subject-suggested terms.

Phase 2: After the initial term solicitation, subjects were asked to use one algorithm and then the other to help them find other topics that might supplement their initial sets of topics. The order of presentation of the two algorithms had previously been randomized for each task. The same experimenter recorded all the selected terms. The same menu-driven interface (as shown in the previous sample session) was used for both algorithms and was operated by the experimenter. For each task, the interface allowed each subject to pick terms suggested by the algorithms, use the newly picked terms to activate the algorithms, pick more terms, activate again, and so on until the subject decided to stop. After using both algorithms, subjects were asked to examine the lists of terms they initially presented and the terms they picked from the system-suggested lists and decide whether they were still relevant (sometimes terms may no longer be considered relevant after a subject has identified other more precise terms).

After using the algorithm-based interface, we asked subjects to browse the three knowledge bases (KBs) for terms that might supplement their initial sets of terms (subject-suggested terms). A simple hypertext-like interface was used to navigate through the three KBs. For each task, each subject examined all three KBs separately, in the subject's own choice of order. Each subject typed in one term at a time using a specific thesaurus. The subject then chose one relevant term (if any occurred) from the thesaurus-suggested list and repeated the same process. In essence, the hypertext-like browsing system only looked up terms which were directly linked to a search term in a chosen knowledge base. On the other hand, the two algorithms performed an optimal (or convergent), multiple-link, multiple-thesaurus search for relevant terms, in contrast to the often laborious manual browsing process that is widely used in library and bibliographic database retrieval settings (Chen and Dhar, 1987). After browsing all KBs, subjects were asked to re-examine the lists of selected terms and to confirm their selections.

Phase 3: After the above term selection process (both algorithmic and manual), I proceeded to a document selection and evaluation phase. Due to difficulty in operationalizing the recall and precision measures (especially recall) (Salton et al., 1994), a document evaluation design similar to the one reported in (Ekmekcioglu et al., 1992) was adopted. Subjects were asked to examine different sets of ranked documents for their relevance to the corresponding dissertation abstract. Four lists of terms, representing subject suggested, branch-and-bound suggested, Hopfield net suggested, and browsing selected terms, were used, respectively, to retrieve the 15 most pertinent abstracts from the Public database. Document ranking was based on the number of terms that were associated with each abstract. In addition, we also randomly generated 15 abstracts that did not match with any of the selected terms. Depending on the number of overlapping abstracts, the number of documents in each collection for each task ranged from 43 to 68. By mixing relevant and irrelevant documents and soliciting subjects' evaluation, we were able to find out which set of terms was most helpful in suggesting relevant documents.

On average, each subject spent about 6.2 hours to complete all six tasks through the above stages: subject's term suggestion, two algorithm activations, manual browsing, and document evaluation. I also logged the complete interactions and recorded subjects' verbal comments. Two sets of performance analysis were conducted: one focused on terms and the other on documents. I report the results below.

4.7.2 Experimental Results: Performance Analysis on Concept Retrieval

The final lists of terms picked by each individual subject (i.e., subject-suggested, algorithm-suggested, and browsing-suggested terms) for each task were used as the *target list* of relevant terms. The *term recall*, *term precision*, *time spent*, *contribution rate*, and *reviewing rate* of the branch-and-bound suggested list (SN), the Hopfield net suggested list (NN), and the manual browsing selected list (MB) were then computed.

Interestingly, there was more agreement between subjects' initial lists than their final lists. Most subjects used the terms in the SIGIR abstracts to start their searches, thus there was a significant overlap of terms used. However, after initiating the manual browsing process and the algorithms, their selections varied significantly. I believe this was because of the differences between the subjects' backgrounds and their interpretations of the SIGIR abstracts.

A. Term Recall:

Term recall indicated the portion of the target list which was found on each of the four lists. Results are shown in Figure 4.3. Manual browsing (MB) resulted in a (statistically) significantly higher recall (at P = 10% level) over the branch-and-bound (SN) and Hopfield net (NN) algorithms (SN vs. NN vs. MB, P = 0.013). Two two-sample t-tests confirmed this finding (SN:MB = 0.34:0.44, P = 0.021; NN:MB = 0.33:0.44, P = 0.014). Between the two algorithms, the branch-and-bound algorithm had a slightly higher recall than the Hopfield net algorithm, but the difference was not significant (SN:NN = 0.34:0.33, P = 0.778). After extensive manual browsing, subjects were able to

```
A.  Term Recall:   ANALYSIS OF VARIANCE
  SOURCE     DF        SS        MS        F        p
  FACTOR      2    0.1224    0.0612     4.71    0.013
                                       INDIVIDUAL 95 PCT CI'S FOR MEAN
   LEVEL      N      MEAN     STDEV   ----+---------+---------+---------+--
  SN         18    0.3421    0.0976     (--------*--------)
  NN         18    0.3326    0.1026   (--------*--------)
  MB         18    0.4380    0.1377                     (--------*--------)
                                      ----+---------+---------+---------+--
  POOLED STDEV =   0.1140               0.300     0.360     0.420     0.480
```

Figure 4.3: Result of term recall on term retrieval

obtain a larger set of relevant terms (than resulted from using the algorithmic process).

B. Term Precision:

Term precision indicated the portion of each list that appeared in the target list. Results are shown in Figure 4.4. In contrast to the recall results, manual browsing resulted in a significantly lower precision value compared with the algorithm-based systems (SN vs. NN vs. MB, P = 0.000). This was mainly because the number of terms suggested via manual browsing was much larger than that of terms suggested by the two algorithms, and only a few were judged relevant by the subjects. Two two-sample t-tests also confirmed this finding (SN:MB = 0.19:0.02, P = 0.000; NN:MB = 0.18:0.02, P = 0.000). Although the branch-and-bound algorithm had a slightly higher precision

```
B.  Term Precision:  ANALYSIS OF VARIANCE
  SOURCE     DF        SS        MS        F        p
  FACTOR      2   0.31097   0.15549    42.53    0.000
                                      INDIVIDUAL 95 PCT CI'S FOR MEAN
   LEVEL      N      MEAN     STDEV   -+---------+---------+---------+-----
  SN         18   0.19271   0.06632                           (----*---)
  NN         18   0.17580   0.08014                         (---*---)
  MB         18   0.02394   0.01207   (---*----)
                                      -+---------+---------+---------+-----
  POOLED STDEV =  0.06046            0.000     0.070     0.140     0.210
```

Figure 4.4: Result of term precision on term retrieval

than the Hopfield net algorithm, the difference was not significant (SN:NN = 0.19:0.18, P = 0.495).

C. Time Spent:

Because the hypertext-like interface suggested many more terms than the two algorithms, the time spent on manual browsing was significantly longer than that spent using either of the two algorithms (SN vs. NN vs. MB, P = 0.000). On average, each subject spent 16.2 minutes browsing, while each spent 5.9 to 7.2 minutes using either algorithm for each task. Results are shown in Figure 4.5. Two two-sample t-tests revealed the same result (SN:MB = 5.9:16.2, P = 0.000; NN:MB = 7.2:16.2, P = 0.000). Between

```
C.  Time Spent:  ANALYSIS OF VARIANCE
  SOURCE     DF        SS        MS        F        p
  FACTOR      2    1121.9     561.0    21.71    0.000
                                      INDIVIDUAL 95 PCT CI'S FOR MEAN
   LEVEL      N      MEAN     STDEV   ---+---------+---------+---------+---
  SN         18     5.944     3.226   (----*----)
  NN         18     7.167     3.468      (---*----)
  MB         18    16.167     7.422                    (---*----)
                                      ---+---------+---------+---------+---
  POOLED STDEV =    5.084              5.0      10.0      15.0      20.0
```

Figure 4.5: Result of time spent on term retrieval

the two algorithmic systems, subjects spent less time using the branch-and-bound system, but the difference was not significant (SN:NN = 5.9:7.2, P = 0.281).

D. Contribution Rate:

I defined a new measure, *contribution rate*, to indicate the number of terms picked by subjects per time unit (one minute). Analysis of variance showed that the contribution rate of manual browsing was significantly lower than that of either algorithm (SN vs. NN vs. MB. P = 0.001). Results are shown in Figure 4.6. On average, the manual browsing process contributed 0.58 term per minute, while branch-and-bound and Hopfield net systems contributed 1.44 and 1.22 terms per minute, respectively. Two two-sample t-tests also confirmed this finding (SN:MB = 1.44:0.58, P = 0.000; NN:MB =

```
D.  Contribution Rate:  ANALYSIS OF VARIANCE
  SOURCE     DF        SS        MS        F        p
  FACTOR      2     7.085     3.542     7.90    0.001
                                      INDIVIDUAL 95 PCT CI'S FOR MEAN
   LEVEL      N      MEAN     STDEV   -----+---------+---------+---------+-
  SN         18    1.4368    0.7907                    (------*-----)
  NN         18    1.2243    0.8263                (-----*------)
  MB         18    0.5846    0.1916   (------*-----)
                                      -----+---------+---------+---------+-
  POOLED STDEV =   0.6695                0.50      1.00      1.50      2.00
```

Figure 4.6: Result of contribution rate on term retrieval

1.22:0.58, P = 0.003). However, the difference between the two algorithms' contribution rates was not significant (SN:NN = 1.44:1.22, P = 0.436).

E. Reviewing Rate:

A subject was expected to spend about the same amount of time reviewing terms suggested by algorithms and terms returned from manual browsing. Nonetheless, in this experiment, the *reviewing rate* (the number of terms reviewed per time unit – one minute in this case) for manual browsing was significantly faster than that for both algorithms (SN vs. NN vs. MB, P = 0.000). Results are shown in Figure 4.7. Individual two-sample t-tests confirmed the result (SN:MB = 7.5:29.6, P = 0.000; NN:MB = 6.9:29.6, P = 0.000). There was no significant difference between the two algorithms (SN:NN = 7.5:6.9, P = 0.565). The hypertext-like browsing process was

```
E.  Reviewing Rate:  ANALYSIS OF VARIANCE
  SOURCE     DF        SS        MS        F        p
  FACTOR      2    6053.7    3026.9    41.72    0.000
                                      INDIVIDUAL 95 PCT CI'S FOR MEAN
   LEVEL      N      MEAN     STDEV   --------+---------+---------+--------
  SN         18     7.477     3.385    (---*----)
  NN         18     6.861     2.960    (---*---)
  MB         18    29.623    14.051                           (---*---)
                                      --------+---------+---------+--------
  POOLED STDEV =    8.518                    10        20        30
```

Figure 4.7: Result of reviewing rate on term retrieval

clearly more time-consuming and cognitively demanding than the algorithmic process.

Using their own terms, subjects were able on average to achieve a 30% recall level. Their precision level was at 100% – all terms they initially supplied were later judged relevant. Each algorithm was able to double the number of terms subjects selected, as shown by the 33% and 34% recall values from the two algorithmic systems. Human ability to *recognize* objects (much better than *recall* of objects) has been well documented in the cognitive psychology literature (Anderson, 1985a), (Chen and Lynch, 1992). The algorithms' suggestions in effect served as an excellent memory-jogging tool for users during concept exploration. The same memory-jogging factor also enabled manual browsing to provide more terms – 44% recall value for manual browsing. However, the time spent and effort involved in manual browsing were enormous for all subjects.

An interesting observation from the experiment, based on the subjects' verbal protocols, was that the manual browsing process often resulted in serendipitous and/or off-the-track browsing behavior as reported in (Carmel et al., 1992). A subject commented, "I just want to have a quick look at this term. It is related to what I am doing (subject's own research) right now. It's not relevant to this query." In addition, manual browsing provided many opportunities for distractions. While reviewing a list of terms, a subject said, "I am curious about this term. Let me see what it is." I observed that when subjects used the algorithms, they reviewed the suggested terms more slowly and treated them more seriously and carefully than when performing manual browsing. Subjects glanced or skimmed through most terms retrieved from manual browsing. Another observation was that subjects were easily frustrated when they reached the end of a particular link during the manual browsing process. A subject said, "It's a dead end. ... It's a dead end, again!" Whereas the physical appearances of all three knowledge bases were left open for subjects to browse and examine, both of our algorithms hid physical appearance from subjects.

In conclusion, manual browsing achieved higher term recall but lower term precision than the algorithmic systems. It was also a much more laborious and cognitively demanding process.

4.7.3 Experimental Results: Performance Analysis on Document Retrieval

For document retrieval performance analysis, I first compared documents obtained from four different sources: subject-suggested (Subject), branch-and-bound algorithm (SN), Hopfield net algorithm (NN), and manual browsing process (MB) (A. and B. of Figure 4.8). I then combined the two sets of algorithm-related documents to determine performance differences among human effort (Subject), algorithmic approach (SN/NN), and manual browsing approach (MB) (C. and D. in Figure 4.9). (The two algorithms suggested different sets of documents which can be combined easily in a system.) Document recall and precision results are discussed below.

A. Document Recall (Subject, SN, NN, MB):

Document recall indicated the portion of the target document list which was found in each of the four lists. The document recall for manual browsing was somewhat higher than that for subject-suggested, branch-and-bound, and Hopfield net algorithms, but the overall analysis of variance did not show a significant difference (Subject:SN:NN:MB = 0.31:0.27:0.29:0.43, P = 0.202). However, the two-sample t-test between the branch-and-bound procedure and the manual browsing process did indicate a significant difference (SN:MB = 0.27:0.43, P = 0.068). Other individual two-sample t-tests did

```
A. Document Recall (Subject, SN, NN, MB):  ANALYSIS OF VARIANCE
  SOURCE     DF        SS        MS        F        p
  FACTOR      3    0.2777    0.0926     1.58    0.202
                                     INDIVIDUAL 95 PCT CI'S FOR MEAN
   LEVEL      N      MEAN     STDEV  -------+---------+---------+---------
  Subject    18    0.3096    0.2347     (---------*--------)
  SN         18    0.2745    0.1934  (---------*--------)
  NN         18    0.2880    0.2339    (--------*--------)
  MB         18    0.4312    0.2944              (---------*--------)
                                     -------+---------+---------+---------
  POOLED STDEV =   0.2418                 0.24      0.36      0.48


B.  Document Precision (Subject, SN, NN, MB):  ANALYSIS OF VARIANCE
  SOURCE     DF        SS        MS        F        p
  FACTOR      3    0.0545    0.0182     0.65    0.583
                                     INDIVIDUAL 95 PCT CI'S FOR MEAN
   LEVEL      N      MEAN     STDEV  -+---------+---------+---------+-----
  Subject    17    0.2107    0.2177            (----------*----------)
  SN         18    0.1481    0.1560   (----------*----------)
  NN         18    0.1444    0.1258  (-----------*----------)
  MB         18    0.1889    0.1572         (----------*----------)
                                     -+---------+---------+---------+-----
  POOLED STDEV =   0.1667           0.070     0.140     0.210     0.280
```

Figure 4.8: Results of independent algorithms on document retrieval

```
C.  Document Recall (Subject, SN/NN, MB):  ANALYSIS OF VARIANCE
  SOURCE     DF        SS        MS        F       p
  FACTOR      2    0.1736    0.0868     1.21   0.305
                                      INDIVIDUAL 95 PCT CI'S FOR MEAN
   LEVEL      N      MEAN     STDEV   -----+---------+---------+---------+-
  Subject    18    0.3096    0.2347   (----------*---------)
  SN/NN      18    0.4286    0.2697             (----------*---------)
  MB         18    0.4312    0.2944             (----------*---------)
                                      -----+---------+---------+---------+-
  POOLED STDEV =   0.2674                  0.24      0.36      0.48      0.60


D.  Document Precision (Subject, SN/NN, MB):  ANALYSIS OF VARIANCE
  SOURCE     DF        SS        MS        F       p
  FACTOR      2    0.0455    0.0227     0.79   0.460
                                      INDIVIDUAL 95 PCT CI'S FOR MEAN
   LEVEL      N      MEAN     STDEV   --+---------+---------+---------+----
  Subject    17    0.2107    0.2177             (-----------*-----------)
  SN/NN      18    0.1405    0.1246    (----------*-----------)
  MB         18    0.1889    0.1572          (-----------*----------)
                                      --+---------+---------+---------+----
  POOLED STDEV =   0.1698             0.070     0.140     0.210     0.280
```

Figure 4.9: Results of combined algorithms on document retrieval

not show significant differences (Subject:SN, P = 0.627; Subject:NN, P = 0.784; Subject:MB, P = 0.180; SN:NN, P = 0.851; NN:MB, P = 0.116).

B. Document Precision (Subject, SN, NN, MB):

Document precision indicated the portion of each document list that appeared in the target document list. The document precision values for subject-suggested and manual browsing sets appeared to be higher than those of the others. However, there was no statistically significant difference among four sources (Subject:SN:NN:MB = 0.21:0.15:0.14:0.19, P = 0.583). Individual two-sample t-tests also did not reveal any significant differences (Subject:SN, P = 0.334; Subject:NN, P = 0.275; Subject:MB, P = 0.735; SN:NN, P = 0.938; SN:MB, P = 0.441; NN:MB, P = 0.356). (Because the terms provided by one subject for a particular task did not retrieve any documents, the sample size for Subject was reduced from 18 to 17 as shown in item B. in Figure 4.8 and item D. in Figure 4.9.)

C. Document Recall (Subject, SN/NN, MB):

The document recall values for the combined algorithms and manual browsing were almost the same. While these values were higher than for subjects, the differences were not statistically significant (Subject:SN/NN:MB = 0.31:0.43:0.43, P = 0.305). Individual two-sample t-tests confirmed the same finding (Subject:SN/NN, P = 0.167; Subject:MB, P = 0.180; SN/NN:MB, P = 0.978).

D. Document Precision (Subject, SN/NN, MB):

The document precision for SN/NN was less than that for Subject and MB. However, there were no significant differences among them (Subject:SN/NN:MB = 0.21:0.14:0.19, P = 0.460). Individual two-sample t-tests confirmed the result (Subject:SN/NN, P = 0.247; Subject:MB, P = 0.735; SN/NN:MB, P = 0.314).

In conclusion, no significant differences (in document recall and precision) were observed between the relevant documents suggested by the algorithms and those generated via the manual browsing process. However, the algorithmic thesaurus consultation approach and the manual thesaurus browsing process each could contribute to a larger set of relevant documents for users (than was discovered without such aids). In light of the effort required for the manual browsing process, our proposed algorithmic approach appeared to be a viable option for efficiently traversing large-scale, multiple thesauri (knowledge networks).

4.8 Discussion and Conclusion

The potentially large amount of knowledge that can be discovered by various AI, statistical, and neural net learning algorithms and the need to integrate different sources of knowledge have made *knowledge management* or *concept exploration* in a large knowledge space an important area for research.

Based on a hybrid network representation of a semantic net and a neural net, I have proposed two paradigms for automatic traversal in heterogeneous knowledge networks: one based on a symbolic, branch-and-bound search algorithm and the other on a Hopfield net parallel relaxation algorithm. The branch-and-bound search tried to find the "best" search path based on some cost computations and a system-maintained priority queue of incomplete paths. The Hopfield net activation, on the other hand, performed parallel activation of neighboring nodes and combined weights from all neighbors until the network reached a convergent state. The design goal was to permit obtaining real-time system performance in large-scale knowledge networks.

I tested these two algorithms in an application where three knowledge sources were used for information retrieval. One knowledge source was created by cluster analysis algorithms and the other two knowledge bases were extracted from external sources, all having network structure. The complete knowledge network consisted of 13,617 nodes and 80,106 links. The two algorithms and the knowledge sources were implemented on a DECStation 5000/120.

Two experiments were conducted in an attempt to reveal the performance levels of the two methods and their novel characteristics in comparison to the the conventional hypertext-like browsing process. The benchmark testing used 30 sample

1-term, 2-term, 3-term, 4-term, 5-term, and 10-term queries to determine the computation times and the activation patterns of the various knowledge sources using the two algorithms. The user evaluation experiment allowed three graduate student subjects to interact with the two algorithms for six tasks of concept exploration and document retrieval.

In the benchmark testing, shown in Appendix A, the branch-and-bound algorithm was faster than the Hopfield net activation. The average branch-and-bound search took about 6.9 seconds and the Hopfield net activation took 24.5 seconds. However, the parallel relaxation process of the Hopfield net appeared to have helped activate multiple thesauri better than the serial branch-and-bound search, regardless of the source of the initial concepts.

The user evaluation experiment revealed that manual browsing achieved higher term recall but lower term precision compared with the algorithmic systems. However, it was also a much more laborious and cognitively demanding process. In document retrieval, there were no statistically significant differences in document recall and precision between the algorithms and the manual browsing process. In light of the effort required to accomplish the manual browsing process, our proposed algorithmic approach appeared to be a viable option for efficiently traversing large-scale, multiple thesauri (knowledge network).

In conclusion, I believe this research has provided insights concerning development of robust and "intelligent" network-based knowledge management and inferencing systems. A sample application of the proposed design could be for "automatic thesaurus consultation" of multiple, heterogeneous thesauri, created either manually or automatically. An algorithmic approach to concept exploration in a large knowledge network can be performed under a real-time computation constraint and could be very useful for interactive, large-scale document retrieval applications. In addition, I believe emergence of the concept exploration process has transformed the static nature of knowledge in information retrieval systems to serve the dynamic nature of user information need more flexibly.

CHAPTER 5

LARGE-SCALE CONCEPT SPACE GENERATION

5.1 Objectives

Chapter 4 demonstrates the ability of the concept exploration process to help users elaborate information needs by using various manual and automatically generally knowledge sources. This chapter focuses on how large domain-specific knowledge sources (concept spaces) can be generated automatically and investigates their quality compared with that of corresponding domain-specific man-made thesauri. In addition, the chapter addresses the feasibility and scalability issues of large-scale concept space generation in terms of domain knowledge and computing resources.

Concept space generation has been the semantic research component for the Illinois Digital Library Initiative (DLI) project, entitled: "Building the Interspace: Digital Library Infrastructure for a University Engineering Community," one of six projects recently funded by NSF/ARPA/NASA. The goal of the project is to transform the Internet into the *Interspace*, in particular by bringing professional and "intelligent" search and display of structured documents to the Net. To address the scalability issue arising from the continuous growth of digital library repositories, this chapter presents in detail a parallel computing approach to creating concept spaces for semantic retrieval. The Illinois DLI project provides the large-scale testbed for this research.

5.2 Research Questions and Methodology

The specific research questions to be investigated were:

- **Question 1**: With regard to computing scalability, would the technique of computer generation of concept spaces be applicable to very large textual databases?
- **Question 2**: With regard to domain specific knowledge scalability, would concept space generation by technology create satisfactory domain-specific concept associations from corresponding textual databases?
- **Question 3**: How does the quality of concept associations in concept space generated from very large textual databases compare with that of a man-made domain-specific thesaurus?

I used the systems development methodology to iterate the system development cycle using personal-sized Unix workstations, workgroup-sized Unix servers, and national resources of a high performance and parallel computing facility. I also used the experimental design methodology to collect quantitative measures - concept recall and precision.

5.3 Background and Issues

5.3.1 Alleviating the Vocabulary Problem Using Concept Spaces

The *vocabulary (difference) problem* in human-computer interactions has been studied extensively in recent years (Furnas et al., 1987), who found that in spontaneous word choice for objects in five domains, two people favored the same term with less than 20% probability. This fundamental property of language limits the success of various design methodologies for vocabulary-driven interaction. In information science, indexing and search uncertainty have been recognized as the primary sources of information retrieval problems. Previous research (Bates, 1986) has shown that different indexers, well trained in an indexing scheme, might assign index terms for a given document differently. It has also been observed that an indexer might use different terms for the same document at different times (possibly because of learning or the cognitive state of mind at indexing). A high degree of uncertainty with regard to search terms has also been reported: searchers tend to use different terms for the same information sought. Because of the indeterminism involved in indexing and searching, an exact match between the searcher's terms and those of the indexer is unlikely (Chen and Dhar, 1987). This often results in poor recall and precision in search.

The vocabulary problem affects every domain of human knowledge. Based on research over the past few decades, it has become clear to information scientists that development of effective online information retrieval systems must consider the cognitive processes and the vocabulary association characteristics of the users.

5.3.2 Concept Association and Thesaurus Work

According to Belkin, users of information retrieval systems bring with them a problem statement which represents an information need. Inherent in all information needs are "anomalous states of knowledge" (ASKs) (Belkin et al., 1982). In a document retrieval system based on ASKs, the searcher's state of knowledge is represented as a network of associations between words. From the structure and characteristics of the network, it is possible to identify anomalies in the state of knowledge. Several models of human memory association wherein knowledge is represented by network-like structures with linked propositions have been suggested. Anderson's work in human memory is particularly pertinent to term associations in retrieval (Anderson, 1985b). According to Anderson, people remember not the exact wording of verbal communication, but the meaning underlying it. The smallest unit of knowledge that can stand as an assertion bearing meaning is the proposition. Memory, then, is represented as a network of such propositions. The strength of the association paths leading to that piece of information contributes to the level of activation being spread. This theory of *spreading activation* has influenced the design of many semantic network based information retrieval systems (Chen et al., 1993).

Many research groups have created vocabulary-based search aids for online information retrieval systems by making use of existing thesauri or dictionaries. Thesauri, in particular, exhibit a structure similar to human word-association networks. While these tools are able to provide the searcher with alternate terms to use in searching, they do not overcome the *knowledge acquisition bottleneck*: the cognitive demand required of humans (indexers or domain experts) to create thesauri or dictionaries in the first place. An alternative approach to creating vocabulary-based search aids is based on *automatic thesaurus generation*.

- **Incorporating existing thesauri:**

  Fox et al. focused on creation of so-called "relational thesauri." For example, CODER adopted the *Handbook of Artificial Intelligence* and *Collin's Dictionary* (Fox et al., 1988). Ahlswede and Evens parsed (Ahlswede and Evens, 1988) *Webster's Seventh New Collegiate Dictionary* to obtain a "lexical database" containing lexical or lexical-semantic relationships from the dictionary definitions. Lesk converted an online version of Murray's *Oxford English Dictionary* into a thesaurus-like tool to facilitate searching of historical manuscripts. These approaches represent attempts to produce "universal lexicons," rather than domain-specific thesauri or dictionaries. Chen et al. conducted a series of experiments which included several large-scale, domain-specific thesauri. In (Chen and Dhar, 1991), Chen and Dhar incorporated a portion of the *Library of Congress Subject Headings* (LCSH) in the computing area into a system that used a branch-and-bound spreading activation algorithm to assist users in query formulation. More recently, they developed concept-based document retrieval using multiple thesauri: two existing thesauri (LCSH and the ACM Computing Review Classification System) and an automatically-generated computing-specific thesaurus (Chen et al., 1993). The National Library of Medicine's *Unified Medical Language System (UMLS)* project is probably the largest-scale effort adopting existing domain-specific knowledge sources or thesauri in information access. It aims to build an intelligent automated system that understands biomedical terms and their interrelationships and uses this understanding to help users retrieve and organize information from multiple online sources (Lindberg and Humphreys, 1990).

- **Automatic thesaurus generation:**

  Numerous investigators have developed algorithmic approaches to *automatic thesaurus generation.* Most of these approaches employ techniques that compute coefficients of "relatedness" between terms using statistical co-occurrence algorithms (e.g., cosine, Jaccard, Dice similarity functions) (Chen and Lynch, 1992), (Salton, 1988), (Rasmussen, 1992). Some algorithms, however, perform cluster analysis to further group terms of similar meanings (Rasmussen, 1992). Other algorithms, such as latent semantic indexing (Dumais, 1994), perform statistical analysis to identify important semantic descriptors. Stiles (Stiles, 1961) was one of the early researchers who reported improved retrieval performance using a method based on term association (with collections of librarian-applied subject tags). Doyle (Doyle, 1962) further argued that the principles underlying association-based retrieval should apply whether the associations are determined by humans or by machines (programs).

More recently, Crouch and Yang (Crouch and Yang, 1992) automatically generated thesaurus classes from text keywords, which can subsequently be used to index documents and queries. Crouch's approach is based on Salton's vector space model and the term discrimination theory.

5.4 Concept Space Techniques

Based on the human information processing theory (Newell and Simon, 1972), (Card et al., 1983), (Anderson, 1985a), Chen et al. argued that creating robust and useful domain-specific thesauri (not universal thesauri) automatically requires a clear understanding of the following seven system development principles: *logarithmic vocabulary growth, completeness, term specificity, asymmetric association, relevance feedback, vocabulary overlapping, and spreading activation* (Chen et al., 1997) Selected algorithms for automatic thesaurus generation have been developed based on these seven principles. The specific steps and algorithms adopted include: *document and object list collection*, *object filtering*, *automatic indexing*, *co-occurrence analysis*, and *associative retrieval.*

The following presents a brief overview of these techniques in the context of this research experiment. For algorithmic details, readers are referred to (Chen and Lynch, 1992), (Chen et al., 1993), (Chen et al., 1995).

5.4.1 Document and object list collection

In any automatic thesaurus building effort, the first task is to identify complete and recent collections of documents in specific subject domains that can serve as the sources of vocabularies. The proliferation of Internet services and the availability of online bibliographic databases have made document collection much easier.

In (Bates, 1986), Bates proposed a design model for subject access in online catalogs. She stressed the importance of building domain-specific lexicons for online retrieval purposes. A domain-specific, controlled list of keywords can help identify legitimate search vocabularies and help searchers "dock" on to the retrieval system. For most domain-specific databases, there appear always to be some existing lists of subject descriptors (e.g., the subject indexes at the back of a textbook), researchers' names (e.g., author indexes or researchers' directories), and other domain-specific objects (e.g., genes, experimental methods, organizational names, etc.) which exist online or can be obtained through OCR scanning.

5.4.2 Object Filtering

For each online document, terms are first identified and then matched with terms in known vocabularies, a process referred to as *object filtering*. Because the texts remaining after object filtering may still contain many important concepts, an automatic indexing procedure, which includes dictionary look-up, stop-wording, word stemming, and term phrase formation, then follows (Salton, 1988).

5.4.3 Automatic Indexing

Automatic indexing begins with stop-wording and continues with term phrase formation. The algorithm first identifies individual words and non-word tokens from free text. Next, a stop-word list is used to remove non-semantic bearing words (e.g., "the", "a", "on", "in"). After removing stop words, a similar procedure is adopted to remove verbs and adverbs. Finally, term phrase formation is accomplished by combining adjacent words. This approach is different from Salton's (Salton, 1988), which forms term phrases with non-stop words from various word positions in a sentence. The main objective of the process of *automatic indexing* is to extract useful *concepts* (*semantics*) from unstructured (free) text of document records.

5.4.4 Co-occurrence Analysis

After terms have been identified in each document, The analysis process computes the term frequency and the document frequency for each term in a document. Term frequency, tf_{ij}, represents the number of occurrences of term j in document i. Document frequency, df_j, represents the number of documents in a collection of n documents in which term j occurs. A few changes in the standard *term frequency* and *inverse document frequency* measures have been made based on the experience gained from the system development process.

Usually terms identified from the title of a document are more descriptive than terms identified from the abstract of the document. In addition, terms identified by the object filters are usually more accurate than terms generated by automatic indexing. Adjustment in determining the weights of importance can be made according to the requirements and needs of different domains and applications.

The combined weight of term j in document i, d_{ij}, is computed based on the product of "term frequency" and "inverse document frequency" as follows:

$$d_{ij} = tf_{ij} \times \log(\frac{N}{df_j} \times w_j)$$

where N represents the total number of documents in a collection and w_j represents the number of words in descriptor j. Multiple-word terms are assigned heavier weights than single-word terms because multiple-word terms usually convey more precise semantic meaning than single-word terms.

Term co-occurrence analysis is performed based on the asymmetric "Cluster Function" developed by Chen and Lynch (Chen and Lynch, 1992).

$$W_{jk} = \frac{\sum_{i=1}^{n} d_{ijk}}{\sum_{i=1}^{n} d_{ij}} \times WeightingFactor(k)$$

$$W_{kj} = \frac{\sum_{i=1}^{n} d_{ikj}}{\sum_{i=1}^{n} d_{ik}} \times WeightingFactor(j)$$

W_{jk} indicates the similarity weights from term j to term k and W_{kj} indicates the similarity weights from term k to term j. d_{ij} and d_{ik} were calculated based on the equation in the previous step. d_{ijk} and d_{ikj} represent the combined weight of both descriptors j and k in document i. However, they are computed slightly differently due to their different starting terms. They are defined as follows:

$$d_{ijk} = tf_{ijk} \times \log\left(\tfrac{N}{df_{jk}} \times w_j\right)$$

$$d_{ikj} = tf_{ijk} \times \log\left(\tfrac{N}{df_{jk}} \times w_k\right)$$

where tf_{ijk} represents the number of occurrences of both term j and term k in document i (the smaller number of occurrences between the terms is chosen). df_{jk} represents the number of documents (in a collection of N documents) in which terms j and k occur together. w_j represents the number of words of descriptor j and w_k represents the number of words of descriptor k (thus descriptors with multiple words receive higher weights).

In order to *penalize* general terms (terms which appeared in many places) in the co-occurrence analysis, the following weighting scheme, which is similar to the *inverse document frequency* function, is applied:

$$WeightingFactor(k) = \frac{\log \frac{N}{df_k}}{\log N}$$

$$WeightingFactor(j) = \frac{\log \frac{N}{df_j}}{\log N}$$

Terms with a higher df_k or df_j value (more general terms) have a smaller weighting factor value, which causes the co-occurrence probability to become smaller. In effect, general terms are *pushed* down in the co-occurrence table (terms in the co-occurrence table are presented in reverse probabilistic order, with more relevant terms appearing first).

5.4.5 Associative Retrieval

In addition to the user-controlled thesaurus browsing process, searchers can also invoke selected spreading activation algorithms for multiple-term, multiple-link term suggestions. The previous chapter presents in detail the use of the serial branch-and-bound algorithm and the parallel Hopfield net algorithm for the spreading activation process. The Hopfield algorithm, in particular, has been shown to be ideal for concept-based information retrieval.

5.5 **Parallel Computing Approach**

Over the past decade, use of parallel computing for information retrieval gradually has progressed from active research to commercial applications (Rasmussen, 1991). Many new classes of algorithms and applications have emerged and created unique opportunities and challenges, especially in the context of Grand Challenge Applications and National Information Infrastructures (Wah, 1993).

5.5.1 Parallel Computing for Information Retrieval

Parallel computing is defined as information processing that emphasizes concurrent manipulation of data belonging to one or more processes solving a single problem. Two classes of architecture have been used to distinguish between different parallel supercomputers: SIMD (single instruction stream, multiple data stream) and MIMD (multiple instruction stream, multiple data stream). In SIMD machines (e.g., MasPar MP-1), one control processor broadcasts a single instruction stream to all the other processors simultaneously for execution on different data streams. In MIMD machines (e.g., Thinking Machine's CM-5), each processor has its own independent program instruction stream. While this classification has been useful in distinguishing between supercomputing architectures, many other classification schemes have recently been proposed.

In (Rasmussen, 1991), Rasmussen suggested three approaches to parallel computing in information retrieval (IR): development and testing of parallel IR algorithms, design of special-purpose parallel hardware for IR applications (e.g., database machines), and development of distributed systems for database access. The first approach, in particular, is of most relevance to this research.

A major focus of research in the 1980s was on the adaptation and refinement of existing popular IR algorithms to parallel processors. Algorithms based on pattern matching (string matching) algorithms or text signatures (superimposed coding) and inverted index file algorithms have attracted most attention (Rasmussen, 1991). Text signatures provide an efficient fixed-length document representation which is ideal for parallel processors (Stanfill and Thau, 1991). However, Salton and Buckley (Salton and Buckley, 1988) have shown that the limited memory units attached to the small processing units on Connection Machine cannot accommodate sophisticated term weights, resulting in significantly degraded performance. In (Couvreur et al., 1994), Couvreur et al. reported the results of modeling the performance of searching large text databases (10+ GBs of *Chemical Abstracts*) via various parallel hardware architectures and search algorithms. They found that a multiprocessor mainframe with parallel inverted index file algorithms and the TRW Fast Data Finder (FDF, special-purpose parallel IR hardware) with "on-the-fly" pattern matching capability out-performed loosely-coupled RISC processors with a text signature algorithm.

While parallelizing existing IR algorithms accounted for a majority of previous research efforts, a significant number of diverse and promising information processing and analysis algorithms have emerged and are believed to be ideal for parallel computation. Co-occurrence analysis and clustering algorithms have traditionally been the most computationally intensive algorithms in information science. As discussed previously, this class of algorithms often aims to compute "relationships" between term-pairs and/or document-pairs and cluster terms/documents of similar nature. As Rasmussen commented (Rasmussen, 1991), "An IR application that is particularly computationally intensive (usually $O(N^2)$ to $O(N^3)$, while offering a high degree of parallelism, is document clustering." He cautioned that the successful implementation of a parallel solution in IR requires an appropriate match of task, algorithm, and architecture.

5.5.2 Parallel Computing for Knowledge Discovery

Many new algorithms developed in the area of machine learning, in particular neural networks and genetic algorithms, are parallel in nature and have become prime candidates for parallel computation. Unlike the IR tasks performed by the previous pattern matching, signature files, and inverted index algorithms, most of these machine learning algorithms require extensive pre-processing and analysis of a significant number of textual documents. (Chen provides a complete and up-to-date review and discussion of machine learning techniques for IR in (Chen, 1995b).)

Neural networks computing, in particular, seems to fit well with conventional retrieval models such as the vector space model (Salton, 1988) and the probabilistic model (Maron and Kuhns, 1960). Oddy and Balakrishnan (Oddy and Balakrishnan, 1991) described a parallel IR system in which a document collection is represented as a network mapped over a Connection Machine. The *PThomas* system is similar to the neural networks model conceptually. However, these researchers noted the practical problem of CM size limits (32K processors), which may render the approach to be infeasible for large-scale databases. They suggested a network partition approach to solving this problem.

Yang and his coworkers (Yang and Korfhage, 1993) have developed adaptive retrieval methods based on genetic algorithms and the vector space model using relevance feedback. They reported the effect of adopting genetic algorithms in large databases, the impact of genetic operators, and GA's parallel searching capability. Frieder and Siegelmann (Frieder and Siegelmann, 1991) also reported a data placement strategy for parallel information retrieval systems using a genetic algorithms approach. Their results compared favorably with pseudo-optimal document allocations.

This emerging machine learning and information analysis paradigm for IR was also echoed in the recent "Report on Workshop on High Performance Computing and Communications for Grand Challenge Applications: Computer Vision, Speech and Natural Language Processing, and Artificial Intelligence" (Wah, 1993). Automatic analysis of co-occurrence patterns in a text corpus, development of electronic librarians to locate information, and discovering new knowledge from existing sources are among the main areas of future research identified by the workshop participants. As the report stated: "High performance AI systems will undoubtedly require very large knowledge bases. Today, the construction of even small to medium knowledge bases is very time-consuming and often prevents the application of AI to real-world problems. To overcome this deficiency, automation of knowledge-base construction is needed. Knowledge may be acquired from the vast amount of information stored as texts. Patterns of concepts and their semantic properties may then be extracted from text via natural language parsing and learning techniques." The research work in the areas of automatic thesaurus generation, spreading activation, and machine learning based IR is a good example of adopting this new "knowledge discovery" paradigm for digital libraries.

5.6 Parellel Computing: Concept Space Generation

In addition to discussing supercomputers based on the SIMD and MIMD architectures, some researchers also classify supercomputers in terms of their processor and memory requirements. According to Larry Smarr, director of NCSA, the first era of supercomputing belongs mainly to "shared memory vector processors" such as Cray X-MP, Cray Y-MP, Cray 2, and Convex C3. The second era of "distributed memory systems" include systems like CM-2, CM-5, IBM clusters, HP clusters, etc. More recently, "shared memory multiprocessors" (SMP) have emerged as the dominant force for the third and current era, e.g., SGI Power Challenge, Convex Exemplar, etc. The parallel computation implementation of the Illinois DLI semantic retrieval research has coincided with the availability of supercomputers of the second and third eras.

As one of the Artificial Intelligent (AI) Lab members at the University of Arizona I have had an opportunity to use all NCSA's supercomputing facility. The rest of this section describes a series of parallel computing implementations and experiments on concept space generation that we have performed from 1994 to 1999. The parallel implementation since Fall 1994 has been on a 512-node CM-5, a 16-processor SGI Power Challenge, a 48-processor SGI Power Challenge Array

(SGIs), a 16-processor Cray CS6400 (SPARCs), a 64-processor Convex Exemplar (HPs), and a 1024-processor SGI Origin2000.

5.6.1 Worm and Fly Concept Spaces Generation Using CM-5 and SGI Power Challenge

In our previous NSF-funded "Worm Community System" project, we adopted the concept space approach to cross-domain scientific information retrieval. By working closely with worm and fly biologists in the Molecular and Cellular Biology Department at the University of Arizona for about two years, we generated a worm thesaurus in Fall 1993 (Chen et al., 1995) and a fly thesaurus in Summer 1994. Both thesauri were independently tested by the biologists and are available for Internet WWW access at: http://bpaosf.bpa.arizona.edu:8000/cgi-bin/BioQuest.

The resulting worm thesaurus consisted of 7,657 terms and 547,810 links and the fly thesaurus contained 15,626 terms and 750,314 links (after applying various thresholds). Most of these terms were author names or subject descriptors. The document collections were mainly Medline and Biosis abstracts acquired from online sources. Each collection contained about 7,000 abstracts, with 10 MBs of text. It took 50 and 70 minutes, respectively, to generate the two thesauri on a DEC Alpha 2100 workstation (200 MHz, 128-MB RAM). Automatic indexing, which is less computationally intensive, took about 2.5 minutes; while the rest of the computation was for co-occurrence analysis. The resulting thesauri were about the same size as the initial document collections (i.e., 1 : 1 storage overhead).

In order to address the scalability issue for concept space generation for large collections, i.e., GB-scale collections, which are common in scientific and engineering domains (the topic domains of the Illinois DLI project), we proceeded to test all concept space algorithms, in particular those of automatic indexing and co-occurrence analysis on supercomputers at NCSA. Our initial platform was the CM-5.

The 7000+ abstracts of the fly collection were used to generate the same fly thesaurus using a 512-node CM-5. CM-5 was mainly based on a massively parallel architecture, with 512 SPARC Cypress processors on distributed memory. Its total memory size is 16 GBs (32 MBs/node on 512 nodes). In automatic indexing, we took the *data parallel* approach by breaking the data file into 100 documents (each 0.5 MB in size) and assigned each 100-document set to one node. Our existing C code was modified to C*, CM's data parallel programming language. In addition, each node needed to access a 2-MB *object filtering* file (with list of known biology terms). Due to the memory size limit of the CM-5 processing unit (node), the automatic indexing process on the same collection took about 19 minutes. However, the same *data parallel* approach worked very well for co-occurrence analysis. By sorting all unique terms in alphabetical order and partitioning them into 22-by-22 (484) chunks of ordered terms, we were able to assign each chunk (0.25 MB) to one CM-5 node. The small file size and processor scale-up greatly speeded up the processing time from 67 minutes (on the DEC Alpha) to 4 minutes, a 17-fold improvement (according to Thomborson (Thomborson, 1993), a more than 20-fold speedup on supercomputers is unlikely without expensive and time-consuming recoding). Although the testing was encouraging, CM-5 did not appear to be able to alleviate the automatic indexing bottleneck for large-scale test collections.

Further testing was conducted in Spring 1995 on the new NCSA Power Challenge (installed in Fall 1994), which is based on a shared memory multiprocessor architecture. It contains 16 MIPS R8000 processors, with a total shared memory size of 4 GBs. Using the same *data parallel* approach, we fully utilized the 16 processors. The resulting processing time was 24 seconds for automatic indexing and 21.5 minutes for co-occurrence analysis. The shared memory architecture alleviated the automatic indexing bottleneck experienced in CM-5, but the co-occurrence analysis time was longer than that on CM-5. All programs were written in C. Table 1 summarizes the CPU time for the 10-MB fly collection on automatic indexing and co-occurrence analysis, respectively, on a DEC Alpha 2100, 512-node CM-5, and 16-processor SGI Power Challenge.

In summary, due to the (distributed) memory size limit of the CM-5 processing unit (32 MBs/node), the automatic indexing process, which involved a large object filtering file, became the bottleneck. However, the 512-node CM-5 was able to perform co-occurrence analysis (a similarity matrix computation by nature, an

Algorithm/Platform	DEC Alpha 2100	CM-5	SGI Power Challenge
Automatic Indexing	2.5 mins	19 mins	24 secs
Co-occurrence Analysis	67 mins	4 mins	21.5 mins
Total	69.5 mins	23 mins	22 mins

Table 5.1: A benchmark comparison summary on DEC Alpha, CM-5, and SGI Power Challenge

$O(N^2)$ process, where N is the number of terms) efficiently after we adopted a data parallel approach, in which each node received a small block (in size) of the matrix to compute. Due to the large shared memory space (4 GBs), automatic indexing was no longer the bottleneck for SGI Power Challenge. However, co-occurrence analysis remained time-consuming, especially for the specific 16-node NCSA SGI Power Challenge we tested (due to a smaller number of processors and a slower clock rate).

The programming learning curve on SGI Power Challenge was significantly smoother than that on CM-5 (roughly 2 weeks vs. 2 months). After some careful consideration and discussions with researchers and staff at NCSA, SGI Power Challenge appeared most promising because of its ease of programming, large shared memory, and expandability. With the planned addition of the 48-processor SGI Power Challenge Array (and faster processors) and supercomputers of similar architecture (e.g., 64-processor Convex Exemplar, and 16-processor Cray CS6400), at NCSA, SMP-based parallel computers were our choice for further experiments.

Our most recent experiment involved a 24-node Convex Exemplar, also provided by NCSA. The NCSA Convex Exemplar employed in September 1995 was a 3-hypernode SPP-1200 system, with 24 HP PA-RISC 7200 chips (processors), 4 GBs of physical memory, and 88 of GBs disk space with peak performance 240 MFLOPS per processor and 1.9 GFLOPS per hypernode. Given more processors and a higher clock rate per processor (compared with the 16-node SGI Power Challenge), automatic indexing for the same fly collection took 0.39 minutes and co-occurrence analysis took 1.46 minutes, both better performances than those of CM-5 and the 16-node SGI Power Challenge.

For large-scale analysis of textual collections, we believe the shared memory multiprocessors (SMP) such as SGI Power Challenge (the NCSA SGI Power Challenge was also recently upgraded) and Convex Exemplar are extremely promising. In a recent issue of *Science* (Pool, 1995), two U.S. supercomputing center directors have commended SMP highly for its architectural fit with the emerging data mining (knowledge discovery) and digital library applications.

5.6.2 Computer Engineering Concept Space Generation Using SGI Power Challenge

In the Illinois DLI project, we obtained an INSPEC test collection of 400,000+ (computer science and electrical engineering abstracts from the 1992-1994 INSPEC database in Spring 1995. This 2-GB testbed recently was used to generate a computer engineering concept space using the 16-processor SGI Power Challenge. The automatic indexing process for this gigabyte collection (about a 200-fold increase in size over the fly/worm collection) took 1.5 hours. The most computationally intensive co-occurrence analysis took 23 hours. It used about 25% of the available CPU cycles on the NCSA Power Challenge for a three-week period (in fact it was the largest single user of the NCSA supercomputing resources at that time). The computer engineering concept space contained about 270,000 terms and 4,000,000+ links. We estimated that running the same program on our DEC Alpha 2100 workstation for the same INSPEC collection would take about 20-30 days of CPU time. This long turn-around time was considered infeasible, due to the iterative nature of our system development and testing effort and the need for testing other even larger collections, e.g., 5 million Compendex engineering abstracts.

We also obtained an online version of the INSPEC thesaurus, which contains 7,000+ terms in a classification hierarchy (mostly narrower-term, broader-term, related-term relationships). Our initial analysis showed that the computer engineering concept space appears to contain finer-grained and newer concepts (terms) than the INSPEC thesaurus. On the other hand, the INSPEC thesaurus provides a richer classification structure of the conceptual relationships than the concept space because of its symbolic links.

The computer engineering concept space was indexed (using WAIS indexing) and recently placed on WWW as an Internet server at: http://ai.bpa.arizona.edu/cgi-bin/csquest. Figure 5.1 shows the search results (related terms) using this server, called *CSQuest*, with a search term, "context analysis." We recommend that this server be used as an interactive computer term suggester for searching any computer-related bibliographical databases (e.g., INSPEC database) or Internet services (e.g., the CS Technical Report projects).

5.6.3 CancerLit Concept Space Generation Using SGI Origin2000

In a medical informatics project, we obtained a CancerLit collection from National Cancer Institute (NCI) that includes close to a million CancerLit document records from January 1992 to June 1999. This 4-GB testbed also represents close to two-thirds of recent cancer research in the entire CancerLit collection. We have created several CancerLit concept spaces exclusively on SGI Origin2000, a new shared memory supercomputing machine in NCSA, since 1997.

Because of the availability of faster processors than those of SGI Power Challenge, the automatic indexing process took less than an hour of processing time on the SGI Origin2000. We tested various processors using the data parallelization strategy. The operating results showed that the wall-clock time (real time elapsed) decreased linearly proportional to the number of processors being used. That is, if 16 processors were used, the wall-clock time to finish the automatic indexing

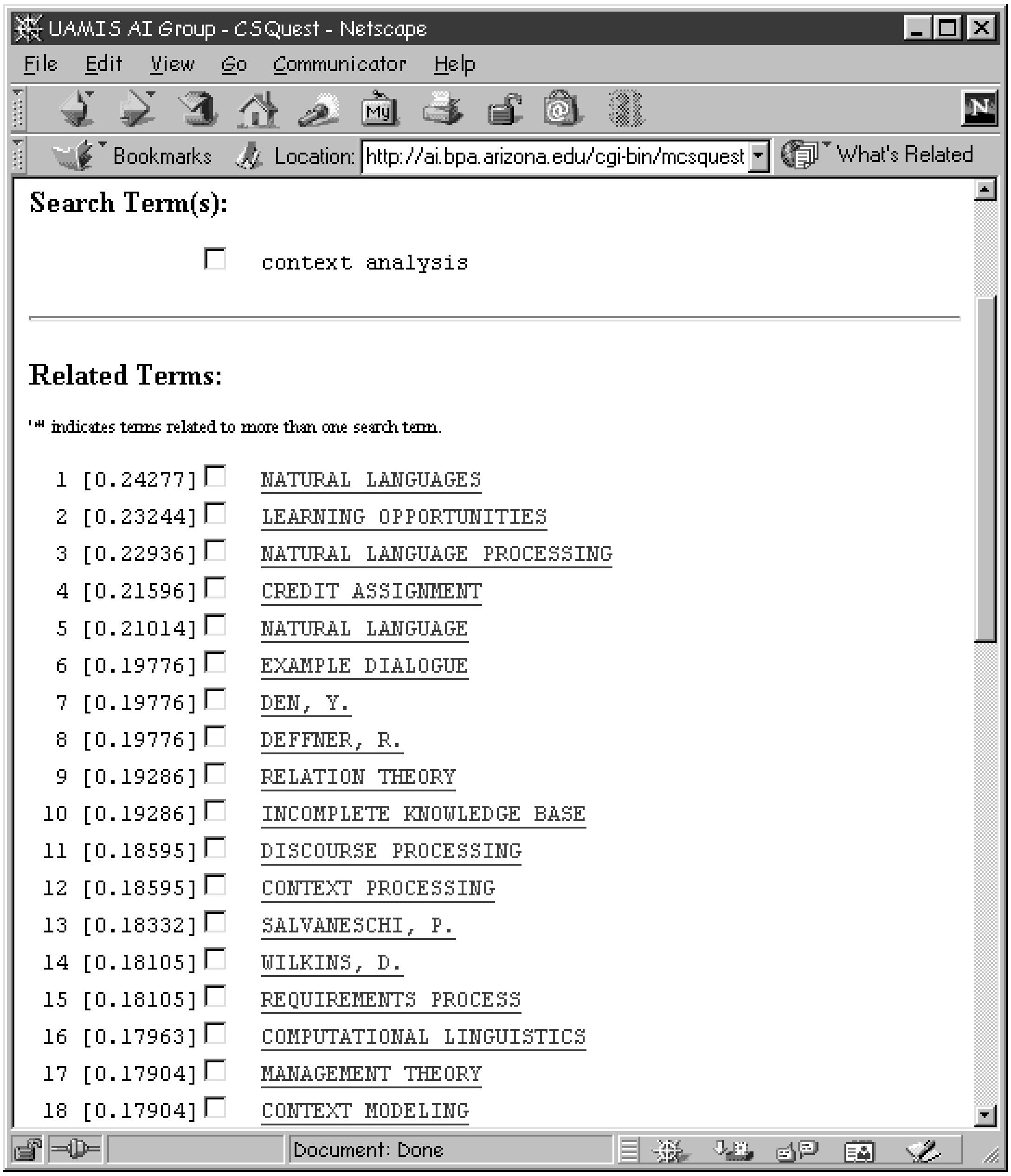

Figure 5.1: CSQuest-suggested terms for “context analysis”

process was about 4 minutes. Memory usage for this automatic indexing process was insignificant. Each processor used about 2 MB of memory to handle input, output and working memory needed. The required lookup memories for stop words and stemming were initialized at less than 0.1 MB of shared memory in less than 1 second of processing time.

Intrinsically, while the data parallelization strategy works well with the automatic indexing process, which is merely an independent analysis on each individual record, parallelization strategy at an algorithmic level is needed for co-occurrence analysis on the entire CancerLit collection. However, we found that a coarse-grained algorithmic parallelization strategy was adequate for co-occurrence analysis when the tremendous memory of up to 64 GB RAM was available and shared simultaneously by multiple processors to perform parallel and independent tasks. We also found that using more processors for co-occurrence analysis might not be beneficial to the overall run-time performance because the use of more processors (up to 32) led to longer total waiting time for peer processes to reach corresponding synchronized points. The coarse-grained strategy allowed longer individual processing cycles between synchronized points to maximize the advantage of having large physical memory. The co-occurrence analysis took 3.5 hours of computing cycles in the SGI Origin2000 running with 8 processors and 4 GB RAM. The improvement in performance came from both better hardware and software optimization.

A revision of CancerLit concept space as of August 1999 contained 1.3 million terms and 27 million links, which covered the underlying CancerLit records from January 1992 to June 1999. A simple CGI-enabled search engine was built to access both CancerLit concept space and document space interchangeably. The server size for the corresponding revision was 3.5 GB. The CancerLit concept space server is accessible at http://ai20.bpa.arizona.edu/cgi-bin/cancerlit/cn.

5.6.4 Lessons Learned: Matching Concept Space Techniques with Parallel Computer Architecture

Both evolution of supercomputer architectures and the maturation of the concept space techniques for very large applications since 1993 have provided tremendous dynamics impetus to understand how to optimize software design and implementation by leveraging on different characteristics of various hardwares and architectures. The strengths and weaknesses of different supercomputer architectures play a significant role in directing the software optimizing effort. Simultaneously, optimization efforts need to fit the nature of concept space techniques at various stages.

Both data and algorithmic parallelization strategies fit well with concept space techniques and are fully supported by the lastest shared memory supercomputer architecture represented in SGI Origin2000. In addition to dicussing these two strategies, the following examines the scalability issues related to the overall concept space generation process.

5.6.4.1 Data Parallelization

Acknowledging that concept space techniques are applicable to information analysis at the single-document as well as the entire-collection level, it nevertheless is natural to separate these two levels of analysis cleanly. The automatic indexing process is single-document analysis while the co-occurrence process is an aggregated analysis at collection level. The data parallelization strategy fits well with the automatic indexing process because the analysis involves taking an input stream, using a small amount of working memory (usually less than 0.1 MB), and producing an output stream. If input data are equally distributed to N processors, a linear reduction in wall-clock run-time will be achieved. We can use as many processors as are available in a computing environment to speed up the automatic indexing process using data parallelization strategy. Indeed, such a strategy will apply equally well on a network of personal computers.

5.6.4.2 Algorithmic Parallelization

The co-occurrence analysis requires simultaneous availability of all relevant data to compute each co-occurring weight between two terms. This requirement demands enormous memory for operation. The shared memory architecture in SGI

Origin2000 fits this requirement. In fact, the size of all necessary data for co-occurrence analysis after pre-processing is under 0.5 GB for a collection of 1 million documents.

The design of the algorithmic parallelization strategy for co-occurrence analysis depends on how much working memory is available and the characteristics of the co-occurrence matrix. Generally, the available memory, other than that for input data, is divided equally among a set of processors for parallel execution. The co-occurrence matrix is very sparse; that is, a term rarely co-occurs with more than 10% of all terms in an analysis. Many terms co-occur only hundreds or thousands of times in a collection. Based on this observation, we used an optimal method to accumulate all co-occurring pairs of terms when processing each document. However, such a co-occurrence process may not have sufficient working memory to hold all pairs of terms. When going through all documents, we therefore mixed a data parallelling strategy to create a portion of term pairs having a different set of starting terms. That is, all documents were parsed multiple times. Using of up to 2 to 4 GB of memory, the number of times to parse all documents was less than 100. Therefore, the performance efficiency of the co-occurrence analysis was directly proportional to the size of memory available to the process. From a series of experiments on executing the co-occurrence analysis with various number of processors ranging from 1 to 32 (not continuously), the increase in number of processors used demonstrated diminishing overall performance efficiency for a given size of memory. This is due to the fact that having more processors reduces the effective size of each one's working memory. The optimal combination of memory and number of processors were found to be 2-4 GB and 8-16 processors, respectively.

5.6.4.3 Synchronization: How Often?

The frequency of synchronization was directly related to whether an algorithmic parallelation strategy applies to a particular process. We chose to use a coarse-grained approach to maximize the ratio of run time to idle time for each processor in the co-occurrence analysis. This also was one of the factors in reaching the point of diminishing return on the performance efficiency when adding more processors. As long as one processor was still working, all other peer processors were forced to wait. The use of more processors simply cumulates more idle time in each synchronization.

5.7 System Evaluation: A Concept Association Experiment

In order to examine the performance of the computer engineering concept space in capturing meaningful conceptual associations between terms, a concept association experiment was conducted using the INSPEC thesaurus as the benchmark for comparison. The experimental design was similar to that of those adopted in memory association (Anderson, 1985a), (Chen and Lynch, 1992) and information retrieval experiments (Chen et al., 1995). The experimental design and results generated are presented in this section.

5.7.1 Experimental Design

A two-phase experiment was performed involving 12 subjects affiliated with the University of Arizona Management Information Systems Department, including two faculty members, and ten graduate degree candidates who had successfully completed course work in Artificial Intelligence, Databases, or Telecommunications/Networking. Prior to Phase 1, experimenters solicited from each faculty subject a list of 16 candidate terms from his domain that could be used as test descriptors. For each domain, eight terms found in both the engineering concept space and the INSPEC thesaurus were selected. One term was discarded because subjects objected to it, leaving a total of 23 test descriptors.

In Phase 1 (*Recall Phase*), each subject (both student and faculty) was asked to generate through a free association process as many related terms as possible in response to each test descriptor presented. This phase of the experiment called upon subjects' memory recall.

In Phase 2 (*Recognition Phase*), experimenters created randomized lists of associated terms for subjects to evaluate with regard to their relevance to the test descriptor, including 40 associated terms suggested by the Computer Engineering Concept Space, and all terms suggested by the INSPEC thesaurus. The concept space offered significantly more than 40 terms; the highest weighted 40 terms were selected for evaluation (about 2 screenfuls of terms). The 12 subjects were then asked to evaluate each suggested term according to the Likert-like scale: "Irrelevant," "Somewhat Relevant," "Very Relevant." Terms considered too general were to be ranked as "Irrelevant." This phase of the experiment called upon the subjects' ability to recognize relevant terms. Human beings are more likely to recognize than to recall. The complete experiment lasted between 1.25 hours and 2.5 hours for each subject.

5.7.2 Experimental Results: Concept Recall and Concept Precision

In contrast to the *document* recall and precision measures typically used in information science research, this experiment adopted *concept recall* and *concept precision* for evaluation. Instead of examining the number of relevant documents retrieved, the number of relevant terms (concepts) identified by the concept space and the INSPEC thesaurus was counted. They were computed as follows:

$$ConceptRecall = \frac{\text{Number of Retrieved Relevant Concepts}}{\text{Number of Total Relevant Concepts}}$$

$$ConceptPrecision = \frac{\text{Number of Retrieved Relevant Concepts}}{\text{Number of Total Retrieved Concepts}}$$

The number of *Retrieved Relevant Concepts* represented the number of concepts judged "Very Relevant" or "Somewhat Relevant" for each thesaurus. *Total Relevant Concepts* represented the target set of concepts that can be obtained through user-thesaurus interaction, and included all concepts generated by the subjects in Phase 1, as well as those additional unique concepts judged relevant from the computer engineering concept space and the INSPEC thesaurus from Phase 2. Graduate student subjects generated between 0 and 49 terms, with a mean of 7.83 terms. Faculty subjects generated between 5 and 30 terms, with a mean of 16.47 terms. Based on this target set of concepts, we examined the relevant concepts generated by each thesaurus to determine the *concept recall*. *Total Retrieved Concepts*, representing the total number of terms suggested by either thesaurus, was used to calculate *concept precision* levels. For the concept space, this value was always 40. The number of retrieved terms offered by the INSPEC thesaurus ranged from 2 to 38, with a mean of 10.391 terms. Two-sample t-tests were performed for *concept recall* and *concept precision*. Separate comparisons were made for each group of subjects (graduate students and experts).

The ten graduate student subjects, responding to each of the 23 test descriptors, generated a total of 230 data sets. The results for *concept recall* and *concept precision* are shown in Figure 5.2. The concept recall results represented a sample size of 218 and included 12 data sets in which subjects did not respond to the test descriptor presented and thus were assigned a *Retrieved Relevant Concepts* value of zero. Concept recall for the automatically generated concept space was 69%, significantly greater than the 17.7% recall value resulting from the INSPEC thesaurus. This result can be attributed to the fact that the concept space offered the subjects a greater number of potentially useful and relevant terms. Of the total set of relevant terms for each test descriptor, approximately 70% came from the automatic concept space. This points to a major advantage of the automatically generated concept space – that it can offer the searcher a richer and more meaningful space for concept association and term suggestion.

Concept precision for the concept space was less than that of the INSPEC thesaurus (59.5% vs. 68.2%), a difference that was statistically significant (at a 10% significance level). That the precision for INSPEC thesaurus was not 100% can be explained by the fact that although terms in a manually generated thesaurus are carefully selected to represent a limited number of highly relevant terms, subjects typically considered broader or parent terms to be irrelevant, which lessened the number of potentially relevant terms within the set suggested. It was not unexpected, then, that the INSPEC thesaurus fared better on precision than did the concept space. Automatic indexing, the technique used in automatic thesaurus generation, not only generates useful, but also noisy terms. Thresholds can be applied to limit this effect, but cannot eliminate it. Therefore the concept space would be expected to contain more potentially irrelevant terms.

Similar results were obtained from the faculty subjects. These subjects responded to those terms relevant to their area of expertise; one responded to Artificial Intelligence terms, the other to Database and Telecommunication/Networking terms. All data sets were completed by the faculty subjects, resulting in a sample size of 23. The experts' concept recall for the automatic thesaurus was somewhat lower than that of student subjects, indicating that the faculty members' criteria for relevance was more stringent than that of the students. In addition, experts tended to have a much higher rate of matching thesaurus and concept space suggested terms. So, while they identified fewer terms as being relevant than did the student subjects, the presence of numerous matching terms from Phase 1 resulted in lower recall. Experts' concept precision was higher than that for the students for both the concept space and the INSPEC thesaurus, primarily because experts failed to respond to far fewer concepts than students. That the difference in precision performance was not statistically significant for the experts is probably attributable to the relatively small sample size.

In conclusion, the automatically generated computer engineering concept space performed much better than the INSPEC thesaurus with regard to concept recall, whereas the INSPEC thesaurus performed better than the concept space with regard to concept precision. The implications of these findings are that the concept space appears to be robust and useful, that the automatically-generated concept space and the manually-created INSPEC thesaurus complement one another, and

```
                                 INDIVIDUAL 95 PCT CI'S FOR MEAN
 LEVEL          N      MEAN     STDEV  -+---------+---------+---------+-----
Rec Auto Stud  218    0.6908    0.1852                                  (*-)
Rec Insp Stud  218    0.1771    0.1381   (*)
                                       -+---------+---------+---------+-----
   POOLED STDEV =     0.1633          0.16      0.32      0.48      0.64

                                 INDIVIDUAL 95 PCT CI'S FOR MEAN
  LEVEL         N      MEAN     STDEV  -+---------+---------+---------+-----
Pre Auto Stud  230    0.5950    0.2822   (--------*--------)
Pre Insp Stud  230    0.6822    0.4153                   (--------*---------)
                                      -+---------+---------+---------+-----
   POOLED STDEV =     0.3551         0.550     0.600     0.650     0.700
```

Figure 5.2: ANOVA analysis for recall and precision with graduate student subjects

```
                                 INDIVIDUAL 95 PCT CI'S FOR MEAN
 LEVEL         N      MEAN     STDEV  ----+---------+---------+---------+--
Rec Auto Fac   23    0.5923    0.1294                              (--*---)
Rec Insp Fac   23    0.1502    0.1064  (--*--)
                                      ----+---------+---------+---------+--
  POOLED STDEV =     0.1185            0.15      0.30      0.45      0.60

                                 INDIVIDUAL 95 PCT CI'S FOR MEAN
 LEVEL         N      MEAN     STDEV  --------+---------+---------+--------
Pre Auto Fac   23    0.6772    0.2004  (-------------*-------------)
Pre Insp Fac   23    0.7366    0.2665          (-------------*-------------)
                                      --------+---------+---------+--------
POOLED STDEV =       0.2358                 0.630     0.700     0.770
```

Figure 5.3: ANOVA analysis for recall and precision with faculty subjects

that the greatest assistance to the searcher can be provided when both are available for query enhancement.

Although the concept association experimental result was encouraging, the usefulness of the concept space for improving user query results (i.e., document recall and precision) was not tested. In a recent AI Lab experiment involving worm (nematode) and fly (Drosophila) concept spaces and a worm database, we have shown that concept spaces helped improve the document recall level (from 32% to 65%), although the document precision level did not improve significantly (Chen et al., 1997). As the next step of our project, we plan to incorporate the engineering concept space in the Illinois engineering literature testbed and examine its usefulness in improving document recall and precision. Future experiments will also include several spreading activation techniques for consulting concept spaces algorithmically for document retrieval, as shown in previous chapter.

5.8 Discussion and Conclusion

This research presents preliminary results generated from the semantic retrieval research component of the NSF/ARPA/NASA-funded Illinois DLI project. Using a variation of the automatic thesaurus generation technique - the *concept space* approach, the goal was to create graphs of domain-specific concepts (terms) and their weighted co-occurrence relationships for all major engineering domains. Merging these concept spaces and providing traversal paths across different concept spaces could potentially help alleviate the *vocabulary (difference) problem* evident in large-scale information retrieval. The AI Lab previously has successfully adopted such a technique for a smaller molecular biology domain (Worm Community System, with 10+ MBs of document collection) with encouraging results.

In order to address the scalability issue related to large-scale information retrieval and analysis for the current Illinois DLI project, we recently proceeded to experiment using the concept space approach on parallel supercomputers. Our test collection was 2+ GBs of computer science and electrical engineering abstracts and the concept space approach called for extensive textual and statistical analysis (a form of knowledge discovery) based on *automatic indexing* and *co-occurrence analysis* algorithms, both previously tested in the biology domain. Initial testing results using a 512-node CM-5 and a 16-processor SGI Power Challenge were promising. Power Challenge was later selected to automatically create a comprehensive computer engineering concept space of about 270,000 terms and 4,000,000+ links using 24.5 hours of CPU time. The user evaluation involving 12 knowledgable subjects revealed that the automatically-created computer engineering concept space generated significantly higher *concept recall* than the human-generated INSPEC thesaurus (concept space : INSPEC thesaurus = 69.08% : 17.71%). However, the INSPEC was more precise than the automatic concept space (concept space : INSPEC thesaurus = 59.50% : 68.22%). Using the INSPEC thesaurus as the benchmark for comparison, I believe the computer engineering concept space has demonstrated its robustness and potential usefulness for suggesting relevant terms for search. However, multiple interfaces and multiple vocabulary search aids will be necessary for effective concept-based search across multiple large-scale repositories and domains.

CHAPTER 6

CONCLUSIONS

This dissertation composes of two tracks of information technology research and development to help users elaborate their information needs when seeking information. The first track is *concept space consultation*, which emphasizes the algorithmic and iterative use of both man-made and machine-generated knowledge sources to present related concepts to users' queries. The second track is *concept space generation*, which shifts the focus from the user aspect to how such knowledge sources can be automatically generated from large library collections.

6.1 Contributions

6.1.1 Dynamic Use of Knowledge Sources

Knowledge sources are static once generated as described in Chapter 2. However, the concept space consultation process dynamically taps into knowledge sources and returns related concepts to users. In addition, users do not need to comprehend the overall knowledge and its structure of any knowledge source. They only need to concentrate on evaluating what is being revealed.

1. *Bringing knowledge to users*

Even though knowledge sources are widely available to users, users often do not use them because of unfamiliarity with different knowledge representations and structures. The concept space consultation provides a transparent access to multiple knowledge sources and brings related concepts to users.

2. *Alleviating from time-consuming and cognitively demanding browsing process*

   Compared with manual browsing, the algorithmic consultation process derives a relatively small set of closely related concepts from multiple knowledge sources. Therefore, the time and cognitive demand for evaluating the usefulness of each concept are less.

6.1.2 Complementary to Man-made Thesauri

Concept associations resemble human-recognizable semantic associations. Moreover, the concept space technique captures concepts and their relationships with respect to underlying library-specific and domain-specific collection as effectively as do man-made thesauri. However, concept spaces and man-made thesauri have two major differences.

1. A concept space provides a more extensive list of associated concepts than does a man-made thesaurus.

2. A concept space can be generated within days while a man-made thesaurus requires years to build.

In a particular domain, an automatically generated concept space and man-made thesaurus compliment each other to capture associative concepts.

6.1.3 Context-specific Concept Spaces

A concept space provides context-specific information with respect to the underlying library database. For example, Figure 6.1 shows two lists of concepts related to "Artificial Intelligence" in the *computer science* and *cancer research* domains, respectively. Each is a ranked list, with the most closely related concepts placed at the top. Both lists have overlapping concepts such as "expert systems" and "neural networks". However, the list in the computer science domain veers toward the context of "learning systems" and "cognitive systems" while the one in the cancer research domain more closely approaches the context of "decision support system" and "decision trees."

The advantage of having context-specific concept spaces is that users can more readily detect whether knowledge sources contain the information they need. This is especially important for information retrieval. If users sense that an appropriate context exists in a knowledge source, it will be very likely that the relevant information is going to be in the underlying library and eventually can be retrieved. If users do not sense any relevant context in a given knowledge source, they will have sufficient information to decide against perusing the underlying library for information.

```
''Artificial Intelligence''
    in Computer Science Domain          in Cancer Research domain
    ----------------------------------- ---------------------------------------
 1  EXPERT SYSTEMS                      Expert Systems
 2  NEURAL NETS                         Algorithms
 3  KNOWLEDGE BASED SYSTEMS             Artifical Neural Networks
 4  INFERENCE MECHANISMS                Breast Neoplasms
 5  LEARNING SYSTEMS                    Computer Systems
 6  KNOWLEDGE REPRESENTATION            DECISION SUPPORT SYSTEM
 7  NEURAL NETWORKS                     Decision Trees
 8  COGNITIVE SYSTEMS                   Discriminant Analysis
 9  PROBLEM SOLVING                     Fuzzy Logic
10  ARTIFICIAL NEURAL NET               Information Systems
11  FORMAL LOGIC                        Neural Networks
12  ARTIFICIAL INTELLIGENCE TECHNIQUES  Pattern Recognition
13  LOGIC PROGRAMMING                   Reproducibility of Results
14  PATTERN RECOGNITION                 SYSTEMS
15  INTELLIGENCE                        Software Design
16  NATURAL LANGUAGES                   Structure-Activity Relationship
17  ROBOTS                              artificial intelligence techniques
18  EXPERT SYSTEM                       knowledge bases
19  FUZZY LOGIC                         knowledge-based systems
20  IMAGE PROCESSING                    Ann Arbor
21  KNOWLEDGE ACQUISITION               Artificial Intelligence research
22  COMPUTER VISION                     BONE MARROW MORPHOLOGY
23  COGNITIVE SCIENCE                   Bayes Theorem
24  KNOWLEDGE ENGINEERING               Brain Neoplasms
25  SEARCH PROBLEMS                     CANCER PATIENTS
26  ARCHITECTURE                        CARCINOGENICITY IN RODENTS
27  SOFTWARE ENGINEER                   CLINICAL DATABASES
28  SOFTWARE ENGINEERING                COMPUTER EXPERT SYSTEM
29  PHILOSOPHICAL ASPECTS               COMPUTER PROGRAM
30  KNOWLEDGE-BASED SYSTEM              COMPUTER SYSTEM
31  COMPUTER SCIENCE                    COMPUTER VISION
32  DISTRIBUTED ARTIFICIAL INTELLIGENCE CONTROL THEORY
33  AI TECHNIQUE                        Carcinogenicity Tests
34  MACHINE LEARNING                    Cervix Uteri
35  INTELLIGENT SYSTEMS                 Chamness GC
36  GENETIC ALGORITHMS                  Clark GM
37  INTEGRATION                         Clinical Protocols
38  MOBILE ROBOTS                       Comparative Study
39  BUILDING                            Computer Automated Structure Evaluation
40  MODELLING                           Computer Graphics
```

Figure 6.1: Context-specific concept space consultation

In addition, users can use contextual information to supplement their background knowledge and shape an expectation of the content of retrieved documents, simply because the contextual information revealed by a concept space comes from a subset of the library - the documents selected for potential retrieval.

6.1.4 Concept Space As Semantic Exchange Medium

The synergy of applying concept space consultation over library-specific concept space goes beyond the function of eliciting user information need. Noting from its experiments and usage, the concept space consultation virtually provides a semantic exchange medium for communicating ideas between users and knowledge sources. Specifically, the exchanging information is the context about the main idea of the "discussion." However, the semantic exchange process in human computer interaction is different from that of human communication.

Contextual information is an essential element for meaningful human communications. Oftentimes, contextual information is hidden in the environment, the background knowledge of communicating parties, and the expectation to the outcome. Nonetheless, our ability to perceive context is often good. Otherwise, we will ask for the context to clarify the main concern. The omission of explicitly expressing contextual (or supportive) information has its pros and cons. The advantage is the simplicity and clarity of staying the main idea of current interest. The disadvantage is that meaningful communication may not be possible because of insufficient or inaccurate understanding of information need - what is really wanted cannot be comprehended by other parties. However, it is not the case in the human-computer communication.

When submitting a query to a knowledge source such as a search engine, a user does not expect the knowledge source to ask about the context of the query. Nevertheless, the use of contextual information is beneficial to help user elaborate information need by having the knowledge source to present its context of user's main idea. The insight of the contextual communication in human-computer communication can be described in the following four steps after a user initiate the communication by querying knowledge source with user's main idea.

1. The knowledge source first acknowledges the recognition of the main concept raised by the user.

2. The knowledge source then presents its knowledge about the context of the main concept.

3. The user is able to perceive how the knowledge source knows about user's main concept.

4. The user may

   (a) stay on the main concept and extend it with other concepts presented by the knowledge source to further explore,

(b) change focus to other suitable concepts because some associated concepts may better describe user information need, or

(c) terminate the exploration because the user does not perceive the knowledge source being able to provide needed information.

Essentially, such a human-computer interaction is a semantic exchange process, which adds the semantic dimension to the user interface to help user find relevant information.

6.2 Lessons Learned

The major lesson learned is that both concept space consultation and generation work. Strictly speaking, machine-generated knowledge sources (concept spaces) are not able to tell what semantics that they are carrying or algorithmic spreading activation process over knowledge sources conveys any semantics. Nonetheless, the computational process in concept space generation captures the knowledge of concepts and co-occurring strengths between concepts from the wisdom in the underlying library collection. Such knowledge does resemble human's knowledge about concepts and their relatively relationships. The computational process in concept space consultation makes use of such human-knowledge-resembling concept associations to further harvest from the derived human wisdom. Both processes are mechanical; however, the overall consultation result is semantical. Concept space consultation provides a semantic exchange medium to help a user engage in expressing an information need - a cognitive process. Observing related concepts found by spreading activation algorithms serves as a memory jogging tool to refine users' articulation of an information need.

From this, a second lesson learned is derived. It is not necessary to present the discovered knowledge in its entirety to users. In fact, doing so may actually scare away potential users because of the tremendous volume of knowledge. Different knowledge sources have their own knowledge representations, with which users are hardly familiar. Besides, the enormous volume of a knowledge source cannot be easily visualized on limited screen real estate. Nonetheless, users can effortlessly benefit from a "strategic" use of knowledge sources - automatic exploration and minimum amount of easy-to-understanding and helpful information.

The third lesson learned is that the automatic concept space technique is scalable conceptually and computationally. The technique works equally well for a wide spectrum of domain areas such as computer science and engineering, medical and health sciences, geosciences and law enforcement, as well as for different sizes of library ranging from hundreds to millions. The technique can be executed on a wide range of platforms including various kinds of Unix and Windows operating systems as well as configurations ranging from mobile laptop computing to high-performance parallel supercomputing.

The final lesson learned is the ability of the consultation process to give an overview of potentially retrieved documents. Such an overview is virtually a summarization method utilizing both context revealed in a concept space and a user's background knowledge. Depending on the information need, such an overview may fit the need, without its being necessary to go through the retrieval process. This lesson is a surprising finding of this dissertation.

6.3 **Future Directions**

In regards to this final finding, I believe there is a promising potential of using concept-oriented other than document-oriented (which clustering often employs) approach to performing summarization over a set of previously conceptually retrieved documents. To the least form of summarization, each document processed by the concept space technique has a ranked list of concepts (term phrases), which can be used as a quick summary for survey. To quantify the value of those *summary lists*, user studies will be needed.

This research demonstrated the feasibility of providing semantic communication between users and knowledge sources or systems. It is logical to extend this semantic communication to creation of a semantic protocol for machine communication. Agents or robots can use the concept space approach to exchange information with context, enabling individual agent or robot to identify whether it is beneficial to communicate with others of its kind, based on the exchanged contextual information. The key to achieve this goal can be inferred from what relevance means in information retrieval. Finding relevant information relies on human judgment on retrieved documents. By default, no agent is able to find relevant information from others using the familiar information retrieval search. The missing piece is the ability to evaluate the relevance of retrieved information. I believe a new search mechanism - contextual search - needs to be developed. The basic idea of contextual search is to represent information need in two categories: the main idea (focus) and contextual (supporting) information.

In this study, concepts are presented in textual format. However, concepts can be in multimedia format. Computationally, concept association works regardless of medium. After concepts can be clearly identified and extracted from multimedia documents, concepts to the process of concept association are merely references. The key of performing multimedia concept association relies on how concepts can be identified and what similarity measure is meaningful in their corresponding media. However, the semantic value that the concept association over multimedia information tries to resemble needs to be determined.

The last item worth noticing is the use of two other types of context analysis. So far, the concept space technique has used *association clusters*, which is the co-occurrence analysis. It may be fruitful to research in the use of *metric clusters* and *scalar clusters* (Baeza-Yates and Ribeiro-Neto, 1999). Metric clusters add the notion of *distance* between co-occurring pair of concepts. However, how *distance* is being measured is an open question. Scalar clusters use the co-occurring (neighboring) concepts of two targeting concepts to compute their similarity. That is, two concepts can be related without being co-occurred in any document in a library.

Appendix A

BENCHMARK TESTING

A.1 Benchmark Testing Design

We first performed a benchmark testing of the two algorithms using 30 sample queries (generated by the experimenters). Each query consisted of terms of various degrees of specificity (e.g., Artificial Intelligence vs. Natural Language Processing) and different numbers of search terms. We tested 5 cases each for queries with 1 term, 2 terms, 3 terms, 4 terms, 5 terms, and 10 terms, a total of 30 cases. A few examples of the queries used, all in the computing area, were: (1-term: Natural Language Processing), (2-terms: Group Decision Support Systems, Collaboration), (3-terms: Systems Analysis and Design, Simulation and Modeling, Optimization), etc.

A.2 Benchmark Testing Results and Discussion

For each query, we selected terms from different knowledge sources, "P" for the Public KB, "A" for the ACM CRCS, and "L" for the LCSH, as shown in Table A.1. Some terms may have appeared in more than one knowledge source. The results shown in Table A.1 reveal the number of iterations, the computing times, and the sources of knowledge for the query terms and the system-suggested terms. It should be note that the branch-and-bound algorithm performed serial iteration, while the Hopfield net performed parallel relaxation at each iteration. The reason for investigating the source of knowledge for system-suggested terms was to show the extent to which each algorithm branched out and utilized knowledge from other knowledge sources.

In response time, the branch-and-bound algorithm clearly performed better than the Hopfield net parallel activation. A MINITAB two-sample t-test (Ryan et al., 1985) showed that on average the neural net took 24.5 seconds (standard deviation, STDEV = 8.34) while the semantic net took 6.9 seconds (STDEV = 2.42). The difference was statistically significant (value of two-sample t-test, T = 11.10 and significance level, P = 0.0000). This was clearly because the branch-and-bound search performed only a fixed number of serial explorations, while the Hopfield net searched a much larger search space during the parallel activation process.

Despite the variation in the number of starting terms, the response times for both methods increased only slightly when the number of starting terms was increased. This finding is important, especially when considering complex, fuzzy queries which often contain many starting terms (a scenario in which searchers need the most help from the system). The reason for this small variation was that our branch-and-bound search decided a threshold based on the user's expected

case	no. of terms	query terms in (P,A,L)	suggested terms in NN:(P,A,L)/SN:(P,A,L)	no. of iterat. NN/SN	times (secs) NN/SN
1	1	(1,1,1)	(12,7,7)/(5,7,2)	18/12	21/11
2	1	(1,0,1)	(5,0,16)/(19,0,2)	15/21	14/8
3	1	(1,1,1)	(11,5,11)/(15,0,0)	14/16	18/10
4	1	(0,0,1)	(0,0,20)/(0,0,20)	11/20	10/11
5	1	(1,0,1)	(4,4,19)/(16,0,3)	17/20	26/10
6	2	(2,1,0)	(19,2,3)/(23,1,0)	21/23	18/6
7	2	(2,0,2)	(16,0,8)/(18,0,1)	19/23	22/8
8	2	(2,0,0)	(20,3,4)/(21,0,0)	20/23	24/5
9	2	(2,1,1)	(11,5,11)/(19,0,0)	15/23	16/4
10	2	(2,1,2)	(11,0,12)/(20,0,0)	27/22	29/4
11	3	(3,0,1)	(20,0,18)/(18,0,0)	19/22	31/5
12	3	(1,2,1)	(4,11,8)/(14,0,2)	22/17	34/6
13	3	(2,1,3)	(22,1,8)/(15,0,2)	18/19	29/6
14	3	(1,3,1)	(20,2,2)/(19,0,0)	16/22	23/8
15	3	(1,2,2)	(13,9,3)/(18,0,1)	9/21	10/4
16	4	(2,2,4)	(17,4,4)/(16,1,1)	17/20	11/6
17	4	(3,2,2)	(11,2,13)/(19,0,1)	19/23	31/5
18	4	(2,3,2)	(18,5,6)/(17,0,2)	24/21	33/4
19	4	(1,3,4)	(18,2,5)/(20,1,1)	19/24	32/7
20	4	(1,2,1)	(15,8,3)/(12,2,1)	18/22	6/7
21	5	(1,4,1)	(19,4,6)/(19,0,1)	16/24	27/3
22	5	(4,2,2)	(10,1,12)/(19,0,1)	15/19	27/4
23	5	(3,2,4)	(2,0,18)/(0,0,21)	11/21	23/9
24	5	(5,0,1)	(19,0,3)/(17,0,1)	23/17	33/9
25	5	(5,0,1)	(20,0,1)/(23,0,0)	12/23	30/12
26	10	(8,0,3)	(11,0,13)/(12,0,19)	17/39	34/6
27	10	(10,1,3)	(13,2,10)/(18,3,2)	25/19	32/7
28	10	(8,0,4)	(16,0,8)/(19,0,2)	24/21	36/8
29	10	(9,1,5)	(19,1,6)/(21,0,1)	27/22	25/9
30	10	(8,2,3)	(20,2,3)/(20,0,2)	28/21	31/6
average	5	(3.1,1.2,1.9)	(14.5,2.5,8.5)/(16.4,0.5,3.0)	18.8/21.3	24.5/6.9

Table A.1: Results of benchmark testing

number of terms. The Hopfield net thresholds (θ_0 and θ_j), on the other hand, were robust enough to guarantee a reasonable number of hits.

Knowledge sources activated by the branch-and-bound algorithm appeared to be more strongly associated with the origins (knowledge sources) of the starting terms than those activated by the Hopfield net. For example, when using the branch-and-bound method, if the starting term was from the LCSH, then the final branch-and-bound suggested terms were more likely to be from the LCSH than from other sources. The Hopfield net, on the other hand, appeared to invoke the different knowledge sources more evenly. As shown in Table A.1, for most queries, the Hopfield net (NN) almost always produced terms from all three knowledge sources (i.e., more evenly), while the branch-and-bound (SN) often produced terms from only a couple of knowledge sources (usually identical to the sources of the query terms). We believe this was because the parallel relaxation process branched out to other knowledge sources more efficiently than the serial search, with the result that combining evidence from different activated nodes as implemented in Hopfield net activation caused more even activation of terms from all sources.

Appendix B

SAMPLE SESSIONS

Sample sessions of branch-and-bound (BAB) and Hopfield net (HP) spreading activations are presented below. Comments are enclosed in parentheses to indicate all the interactions between a subject and the system. The subject was requested to identify topics (with the help of the system) relevant to "KIDS: A Query and Inference System Based upon Knowledge Indexed Deductive Search," by K. Lee, a Georgia State University Ph.D. dissertation, 139 pages, 1989. An abstract of this dissertation was also presented to the subject. Details of the experiment are discussed in the the section on *User Evaluation* in Chapter 4.

```
*-------------------*
    Initial terms:  {* Supplied by the subject and used by both algorithms. *}
    -------------
 1. (P L) INFORMATION RETRIEVAL {* P: Public, A: ACM, L: LCSH *}
 2. (P  ) KNOWLEDGE BASE
 3. (P  ) THESAURUS
 4. (P L) AUTOMATIC INDEXING
*-------------------*
```

A. Branch-and-bound activation:

```
{* The subject selected the branch-and-bound search module first. *}

Enter the number of system-suggested terms or '0' to quit >> 30
{* User supplied his desired number of suggested terms. *}

{* The algorithm searched all three knowledge sources and suggested
```

```
  terms in decreasing order of relevance. Starting terms were included. *}
 1. ( ) THESAURUS
 2. ( ) INDEXING
 3. ( ) KEVIN.HOT              {* User-specific folder in Public DB. *}
 4. ( ) KNOWLEDGE BASE
 5. ( ) INFORMATION RETRIEVAL
 6. ( ) AUTOMATIC INDEXING
 7. ( ) DBMS.AI                {* Terms with *.* are Public folder names. *}
 8. ( ) ROSS.HOT
 9. ( ) INFORMATION RETRIEVAL SYSTEMS
10. ( ) RETRIEVAL
11. ( ) EXPERT SYSTEMS
12. ( ) INFORMATION
13. ( ) DATABASE
14. ( ) CARAT.DAT
15. ( ) QUERY
16. ( ) RECALL
17. ( ) LANGUAGE
18. ( ) SUPPORT
19. ( ) INFORMATION RETRIEVAL SYSTEM EVALUATION
20. ( ) RESEARCH
21. ( ) GQP.DAT
22. ( ) MODEL
23. ( ) KEYWORD
24. ( ) PRECISION
25. ( ) USER INTERFACES
26. ( ) PETER.HOT
27. ( ) ARTIFICIAL INTELLIGENCE
28. ( ) MANAGEMENT
29. ( ) EXPERT SYSTEM
30. ( ) LOGIC
31. ( ) OBJECT
32. ( ) SEMANTIC.MDL
33. ( ) DATABASE MANAGEMENT SYSTEMS
34. ( ) EXPERT
35. ( ) DESIGN

Enter numbers [1 to 35] or '0' to quit:  1, 2, 4-6, 9, 16, 19, 24
{* The subject selected desired terms. *}

{* The system listed the user-selected terms and their sources. *}
 1. (P  ) THESAURUS
 2. (P  ) INDEXING
 3. (P  ) KNOWLEDGE BASE
 4. (P L) INFORMATION RETRIEVAL
 5. (P L) AUTOMATIC INDEXING
 6. (P L) INFORMATION RETRIEVAL SYSTEMS
 7. (P  ) RECALL
 8. (P  ) INFORMATION RETRIEVAL SYSTEM EVALUATION
 9. (P  ) PRECISION
```

```
Enter the number of system-suggested terms or '0' to quit >> 50
{* The subject used the selected terms to activate the branch-and-bound
   algorithm again. *}

{* More terms were suggested. *}
  1. ( ) KEVIN.HOT
  2. ( ) INDEXING
  3. ( ) INFORMATION RETRIEVAL
  4. ( ) RECALL
 ........
 54. ( ) DATA STRUCTURES
 55. ( ) PERFORMANCE
 56. ( ) QUERY.OPT
 57. ( ) ARTIFICIAL INTELLIGENCE
 58. ( ) KEYWORD
 59. ( ) THESAURI
 60. ( ) USER INTERFACES

{* More selections. *}
Enter numbers [1 to 60] or '0' to quit:  2-6, 8, 10, 11, 14, 15
{* Notice that terms were ranked in different order than had
   been suggested previously. This was because of the different
   starting terms. *}

 1. (P  ) INDEXING
 2. (P L) INFORMATION RETRIEVAL
 3. (P  ) RECALL
 4. (P  ) PRECISION
 5. (P L) INFORMATION RETRIEVAL SYSTEMS
 6. (P  ) INFORMATION RETRIEVAL SYSTEM EVALUATION
 7. (P  ) THESAURUS
 8. (P L) AUTOMATIC INDEXING
 9. (  L) INFORMATION STORAGE AND RETRIEVAL SYSTEMS {* A new term. *}
10. (P  ) KNOWLEDGE BASE

Enter the number of system-suggested terms or '0' to quit >> 0
{* The subject decided to stop the search process. *}

{* A total of 10 terms were selected. Six terms were suggested by the
   branch-and-bound algorithm and they came from two knowledge sources:
   the Public KB and the LCSH. *}
```

B. Hopfield net activation:

```
{* The subject selected the Hopfield net search module.
```

```
   Similar interface was used for the Hopfield net search process. *}

Enter the number of system-suggested terms or ‘0’ to quit >> 10
{* Notice that the suggested terms were different from those suggested
   by the branch-and-bound algorithm. *}

  1. ( ) INDEXING
  2. ( ) SELLING - INFORMATION STORAGE AND RETRIEVAL SYSTEMS
  3. ( ) KEVIN.HOT
  4. ( ) INFORMATION RETRIEVAL SYSTEM EVALUATION
  5. ( ) RECALL
  6. ( ) EXPERT SYSTEMS
  7. ( ) THESAURUS
  8. ( ) DBMS.AI
  9. ( ) ROSS.HOT
 10. ( ) INFORMATION STORAGE AND RETRIEVAL SYSTEMS
 11. ( ) INFORMATION RETRIEVAL
 12. ( ) KNOWLEDGE BASE
 13. ( ) AUTOMATIC INDEXING

Enter numbers [1 to 13] or ‘0’ to quit:  1, 2, 4, 5, 7, 10-13

 1. (P  ) INDEXING
 2. (  L) SELLING - INFORMATION STORAGE AND RETRIEVAL SYSTEMS
 3. (P  ) INFORMATION RETRIEVAL SYSTEM EVALUATION
 4. (P  ) RECALL
 5. (P  ) THESAURUS
 6. (  L) INFORMATION STORAGE AND RETRIEVAL SYSTEMS
 7. (P L) INFORMATION RETRIEVAL
 8. (P  ) KNOWLEDGE BASE
 9. (P L) AUTOMATIC INDEXING

Enter the number of system-suggested terms or ‘0’ to quit >> 30
 ........

Enter number [1 to 40] or ‘0’ to quit:  3-7, 9, 33, 35, 36, 38
 ........

Enter numbers [1 to 67] or ‘0’ to quit: 0
{* The system listed his final selections. *}

 1. (P  ) PRECISION
 2. (P L) INFORMATION RETRIEVAL
 3. (P  ) INDEXING
 4. (P L) AUTOMATIC INDEXING
 5. (P  ) RECALL
 6. (  L) AUTOMATIC ABSTRACTING                {* Suggested by HP, not BAB. *}
 7. (  L) AUTOMATIC CLASSIFICATION             {* Suggested by HP, not BAB. *}
 8. (  L) AUTOMATIC INFORMATION RETRIEVAL      {* Suggested by HP, not BAB. *}
 9. (P  ) INFORMATION RETRIEVAL SYSTEM EVALUATION
```

```
10. (P  ) THESAURUS
11. (  L) INFORMATION STORAGE AND RETRIEVAL SYSTEMS
12. (P  ) KNOWLEDGE BASE

{* A total of 12 terms were selected. Eight terms were suggested by the
   Hopfield net algorithm. Terms 6, 7, and 8 were different from those
   suggested by the branch-and-bound algorithm. *}
```

Appendix C

CANCER SPACE: A WEB-BASED INFORMATION RETRIEVAL SYSTEM

Section 5.6.3 describes how a large-scale concept space was generated from approximately one million CancerLit document records. In order to conduct experiments related to research in Digital Library and Medical Informatic as well as provide public access to both CancerLit library and its concept space, a web-based information retrieval system, *Cancer Space*, was built. *Cancer Space* can be found at `http://ai20.bpa.arizona.edu/cgi-bin/cancerlit/cn/`

Although traditional information retrieval systems and web-based search engines have been commercially or freely available, their keyword-based retrieval architectures do not support the needs for indexing and searching concept spaces. A new concept space search engine was designed based on a client-server architecture, which relies on a Common Gateway Interface (CGI) access through a Hypertext Transfer Protocol (HTTP). A web-CGI architecture provides fast system development because of the readily usable network protocol from web servers and graphical user interface from web browsers. During four years of development and use of concept space search engine, various servers have been built on a wide range of platforms ranging from laptops to supercomputers in both Unix and Windows operating systems.

C.1 **Client User Interface**

The client-side implementation provides an interface for users to input their queries and for the corresponding search engine to display search results. The basic design of the user input interface is built on *FORM* elements of Hypertext Markup Language (HTML) (http://www.w3.org/). Initially, users can type in their query terms (one or more), select a *search space*, and submit their queries by clicking on a button.

There are currently two *search spaces*: *Concepts* and *Documents*. Concept search returns *co-occurred terms* from concept space for contextual query expansion, while document search retrieves the actual documents that match the user's query.

- *Concept Space:*

  In concept space search, the interface displays a list of co-occurred terms ranked in a non-decreasing order according to their co-occurred weight in up-to-three columns depending on the types of concept. Currently, the CancerLit concept space carries three types: *Noun Phrase* for terms extracted from free-text, *MeSH* for Medical Subject Headings assigned to each document, and *Author* of documents. The most relevant terms (or concepts) are listed first in their type categories. Users can use the returning list of concepts to help them to further refine or change their search.

Users can select concepts from the co-occurred list and to augment or alter their queries. Users can submit their refined queries to the concept space search and find out what other concepts are closely related to concepts in their query sets. Users can iterate through such a process to find a set of concepts, which best describes their target concept. Such iterative process serves as query refinement to a set of search terms.

Figure C.1 shows a concept space search results from two terms, "Cancer Surgery" and "routine care". For the purpose of backtracking, search terms, which activate concepts appearing in the table, are enumerated by alphabets. For visually emphasizing concepts coming from multiple search terms, color bars are drawn after the first group of having the most number of search terms.

- *Document Space:*

  In document space search, the interface shows a list of relevant documents grouped by the number of matched terms and ranked in a non-decreasing order according to weight calculated by the vector space model. Users have

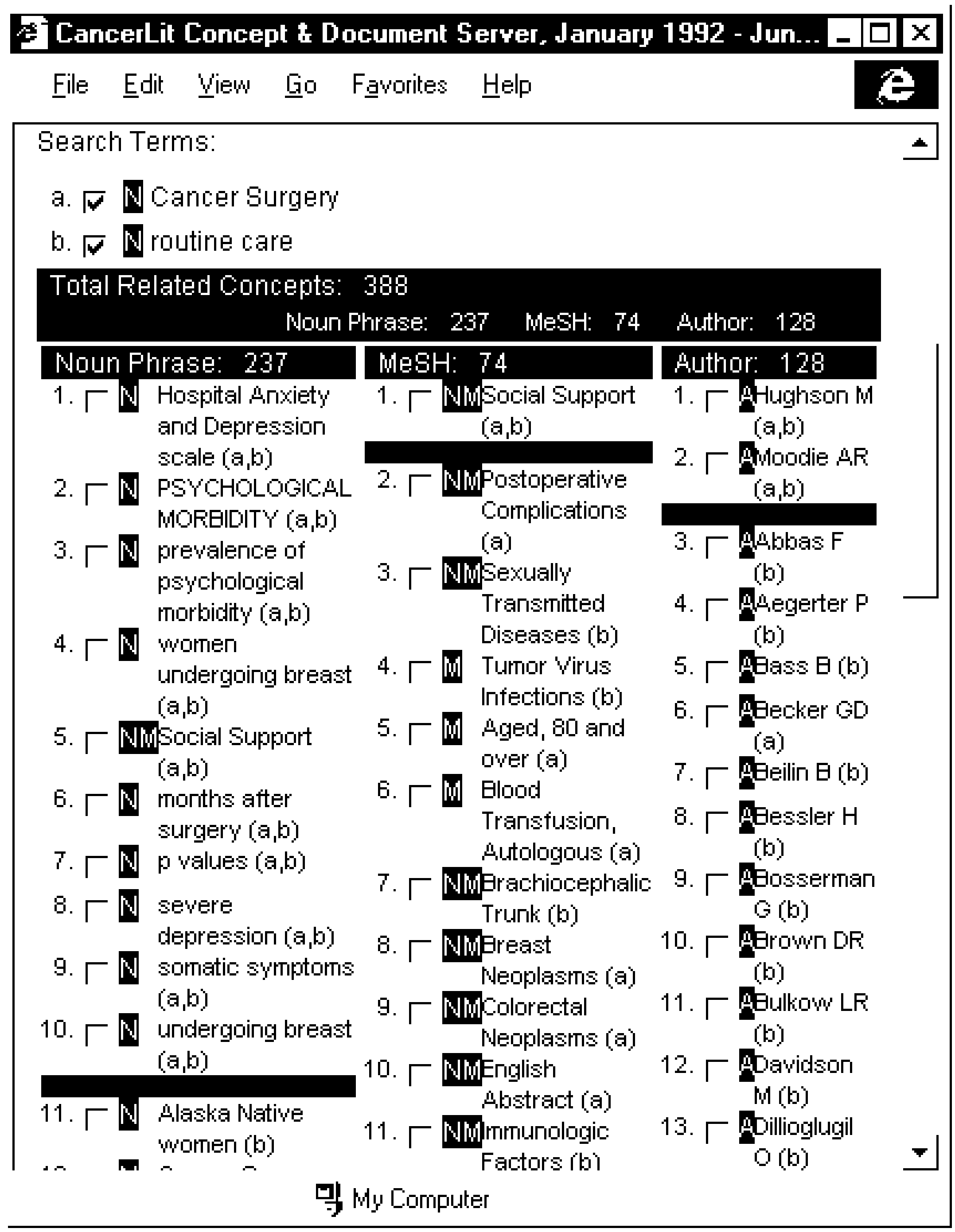

Figure C.1: Cancer Space: Results from concept space search a choice of requesting *Citations* or *Abstracts with Citations* to view document records. The *Citation* shows only *title*, *author*, and *source* of each retrieved document. The *Abstracts with Citations* shows extra information like *MeSH term* assignments and *abstract* to each document.

Figure C.2 shows the document retrieval results from two search terms: "Cancer Surgery" and "routine care". Near the top right corner, it displays document counts for matching both and just one of the two search terms. In this retrieval, two documents have both search terms and ninety-five have just one.

C.2 Server Implementation

The server-side implementation provides a mechanism to store and retrieve information for query input. The storage component is built as a read-only database system, which had several indexing scheme to speed up the retrieval process for all requests. The server uses traditional *keyword* inverted index to index both automatic thesaurus and documents. The difference is that *term phrases* or *concepts* were used instead of single-word keywords as "inverted units" to index their corresponding co-occurred terms and documents.

The retrieval component implemented is similar to most other search engines on the web. The CGI environment provides input (from user query) to a search engine through the HTTP server sitting on our local machine. For both concept

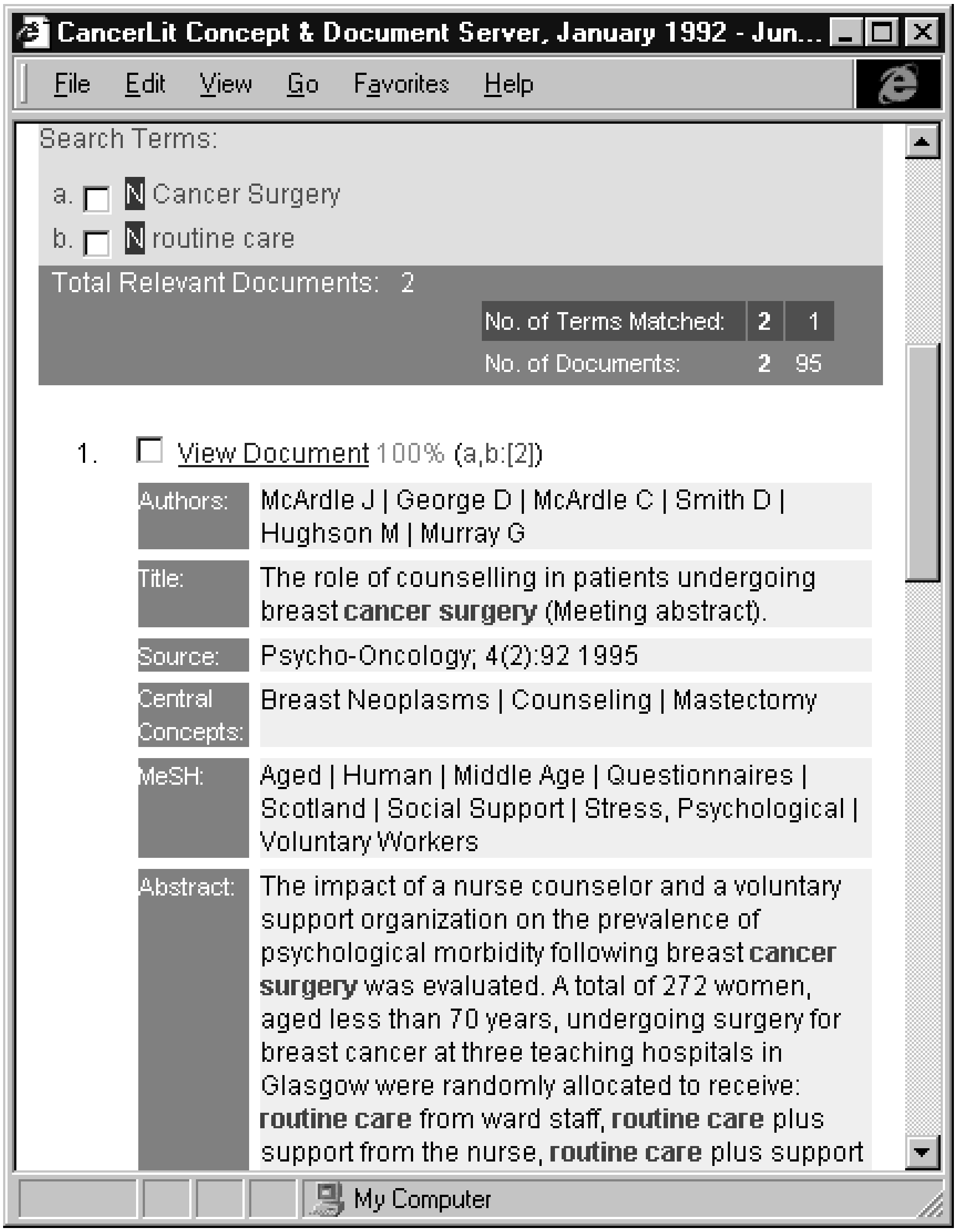

Figure C.2: Cancer Space: Results from document retrieval and document searches, all input terms are used to match against a list of searchable terms in the server. The matching process does some simple stemming and normalization of terms in the same manner in which the thesaurus was generated in order to better connect queries with documents. If at least one input term matches the searchable list, a search of either the concept or document space will be conducted. If no terms match the user's query, the server will issue a message notifying the user of an unsuccessful search. The server formats both results for concept and document spaces in HTML format as dynamic web pages and delivers it to the users' browsers.

Appendix D

FUNDING AND ACKNOWLEDGMENTS

D.1 Concept Space Consultation

This project was supported mainly by a National Institutes of Health (NIH) BRSG grant #S07RR07002 and a National Science Foundation (NSF) grant #IRI-9211418. We would also like to thank K. Basu, K. Ng, C. Wei, J. Martinez, K. Leung, and R. Orwig for their involvement in system development and evaluation.

D.2 Concept Space Generation

This project was supported mainly by the following grants: (1) NSF/ARPA/NASA Digital Library Initiative, IRI-9411318, 1994-1998 (B. Schatz, H. Chen, et al., "Building the Interspace: Digital Library Infrastructure for a University Engineering Community"), (2) NSF CISE Research Initiation Award, IRI-9211418, 1992-1994 (H. Chen, "Building a Concept Space for an Electronic Community System"), (3) NSF CISE Special Initiative on Coordination Theory and Collaboration Technology, IRI-9015407, 1990-1993 (B. Schatz, "Building a National Collaboratory

Testbed"), (4) AT&T Foundation Special Purpose Grants in Science and Engineering, 1994-1995 (H. Chen), and (5) National Center for Supercomputing Applications (NCSA), High-performance Computing Resources Grants, 1994-1996 (H. Chen).

We would also like to thank Dr. Anindya Datta, Pauline Cochrane, and Dr. Ann Bishop for their valuable suggestions, Jim Ashling of the Institution of Electrical Engineers (IEE, vendor of the INSPEC database) for providing the INSPEC thesaurus for our experiment, and Dr. Larry Smarr, Dr. Melanie Loots, and Mike Welge at NCSA for their kind assistance.

REFERENCES

Abecker, A., Bernardi, A., Hinkelmann, K., Kuhn, O., and Sintek, M. (1998). Toward a technology for organizational memories. *IEEE Intelligent Systems*, 13(3):40–48.

Ahlswede, T. and Evens, M. (1988). Generating a relational lexicon from a machine-readable dictionary. *International Journal of Lexicography*, 1(3):214–237.

Anderson, J. R. (1983). *The Architecture of Cognition.* Harvard University Press, Cambridge, MA.

Anderson, J. R. (1985a). *Cognitive Psychology and Its Implications, 2nd Ed.* W. H. Freeman and Company, New York, NY.

Anderson, J. R. (1985b). Indexing systems: extensions of the mind's organizing power. *Information and Behavior,*, 1.

Attar, R. and Fraenkel, A. S. (1977). Local feedback in full-text retrieval systems. *Journal of the ACM*, 24(3):397–417.

Baeza-Yates, R. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval.* Addison-Wesley, Reading, MA.

Bailey, J. A. (1994). Computing support for indexing at petroleum abstracts: design and benefits (online thesaurus). In *Challenges in indexing electronic text and images*. Learned Information.

Barber, J., Moffat, S., and Wood, F. (1988). Cases studies of the indexing and retrieval of pharmacology papers. *Information Processing & Management*, 24(2):141–150.

Bates, M. J. (1979a). Idea tactics. *Journal of the American Society for Information Science*, 30:280–289.

Bates, M. J. (1979b). Information search tactics. *Journal of the American Society for Information Science*, 30:205–214.

Bates, M. J. (1986). Subject access in online catalogs: A design model. *Journal of the American Society for Information Science*, 37(6):357–376.

Bates, M. J. (1990). Where should the person stop and the information search interface start? *Information Processing & Management*, 26(5):575–591.

Bates, M. J. (1998). Indexing and access for digital libraries and the internet: Human, database, and domain factors. *Journal of the American Society for Information Science*, 49(13):1185–1205.

Bechtel, W. and Abrahamsen, A. (1991). *Connectionism and the Mind: An Introduction to Parallel Processing in Networks.* Basil Blackwell, Inc., Cambridge, MA.

Belkin, N. J., Oddy, R. N., and Brooks, H. M. (1982). Ask for information retrieval: Part I. background and theory. *Journal of Documentation*, 38(2):61–71.

Blair, D. C. (1986). Indeterminacy in the subject access to documents. *Information Processing & Management*, 22(2):229–241.

Blair, D. C. (1990). *Language and representation in information retrieval.* Elsevier Science Publishers, New York.

Braine, M. D. S. and O'Brien, D. P. (1998). Logical inferences and comprehension: How mental-logic and text processing theories need each other. In Braine, M. D. S. and O'Brien, D. P., editors, *Mental Logic*, chapter 5, pages 78–90. Lawrence Erlbaum Assocates, Mahwah, NJ.

Bruza, P. D. and Huibers, T. W. C. (1996). A study of aboutness in information retrieval. *Artificial Intelligence Review*, 10:381–407.

Carbonell, J. G., Michalski, R. S., and Mitchell, T. M. (1983). An overview of machine learning. In *Machine Learning, An Artificial Intelligence Approach*, Pages 3-23, Michalski, R. S., Carbonell, J. G., and Mitchell, T. M., Editors, Tioga Publishing Company, Palo Alto, CA.

Card, S. K., MacKinlay, J. D., and Shneiderman, B. (1998). *Readings in Information Visualization: Using Vision to Think.* Morgan Kaufmann Publishers.

Card, S. K., Moran, T. P., and Newell, A. (1983). *The Psychology of Human Computer Interaction*. Lawrence Erlbaum Associates, Hillsdale, NJ.

Carmel, E., Crawford, S., and Chen, H. (1992). Browsing in hypertext: A cognitive study. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(5):865–884.

Chan, L. M. (1989). Inter-indexer consistency in subject cataloging. *Information Technology and Libraries*, 8(4):349–358.

Charniak, E. (1981). A common representation for problem solving and language comprehension information. *Artificial Intelligence*, 16:225–255.

Chen, H. (1994). Collaborative systems: Solving the vocabulary problem. *IEEE Computer*, 27(5):58–66. Special Issue on Computer-Supported Cooperative Work.

Chen, H. (1995a). Machine learning for information retrieval: Neural networks, symbolic learning, and genetic algorithms. *Journal of the American Society for Information Science*, 46(3):194–216.

Chen, H. (1995b). Machine learning for information retrieval: neural networks, symbolic learning, and genetic algorithms. *Journal of the American Society for Information Science*, 46(3):194–216.

Chen, H., Danowitz, A. K., Lynch, K. J., Goodman, S. E., and McHenry, W. K. (1994a). Explaining and alleviating information management indeterminism: A knowledge-based framework. *Information Processing & Management*, 30(4):557–577.

Chen, H. and Dhar, V. (1987). Reducing indeterminism in consultation: a cognitive model of user/librarian interaction. In *Proceedings of the 6th National Conference on Artificial Intelligence (AAAI-87)*, pages 285–289, Seattle, WA.

Chen, H. and Dhar, V. (1991). Cognitive process as a basis for intelligent retrieval systems design. *Information Processing and Management*, 27(5):405–432.

Chen, H., Hsu, P., Orwig, R., Hoopes, L., and Nunamaker, J. (1994b). Automatic concept classfication of text from electronic meetings. *Communications of the ACM*, 37(10):56–73.

Chen, H. and Lynch, K. J. (1992). Automatic construction of networks of concepts characterizing document databases. *IEEE Transactions on Systems, Man and Cybernetics*, 22(5):885–902.

Chen, H., Lynch, K. J., Basu, K., and Ng, D. T. (1993). Generating, integrating, and activating thesauri for concept-based document retrieval. *IEEE EXPERT, Special Series on Artificial Intelligence in Text-based Information Systems*, 8(2):25–34.

Chen, H., Martinez, J., Ng, T. D., and Schatz, B. R. (1997). A concept space approach to addressing the vocabulary problem in scientific information retrieval: an experiment on the Worm Community System. *Journal of the American Society for Information Science*, 48(1):17–31.

Chen, H., Schatz, B. R., Yim, T., and Fye, D. (1995). Automatic thesaurus generation for an electronic community system. *Journal of the American Society for Information Science*, 46(3):175–193.

Cohen, P. R. and Kjeldsen, R. (1987). Information retrieval by constrained spreading activation in semantic networks. *Information Processing and Management*, 23(4):255–268.

Cooper, W. S. (1969). Is interindexer consistency a hobgoblin? *American Documentation*, 20:268–278.

Cooper, W. S. (1971). A definition of relevance for information retrieval. *Information Storage and Retrieval*, 7:19–37.

Couvreur, T. R., Benzel, R. N., Miller, S. F., and Zeitler, D. N. (1994). An analysis of performance and cost factors in searching large text databases using parallel search systems. *Journal of the American Society for Information Science*, 45(7):443–464.

Crouch, C. J. and Yang, B. (1992). Experiments in automatic statistical thesaurus construction. In *Proceedings of the Fifteenth Annual International*

ACM/SIGIR Conference on Research and Development in Information Retrieval, pages 77–88, Copenhagen, Denmark.

Dalton, J. and Deshmane, A. (1991). Artificial neural networks. *IEEE Potentials*, 10(2):33–36.

Derthick, M. (1988). *Mundane Reasoning by Parallel Constraint Satisfaction.* Carnegie Mellon University, Pittsburgh, PA, Ph.D. thesis.

Dietterich, T. G. and Michalski, R. S. (1983). A comparative review of selected methods for learning from examples. In *Machine Learning, An Artificial Intelligence Approach*, Pages 41-81, Michalski, R. S., Carbonell, J. G., and Mitchell, T. M., Editors, Tioga Publishing Company, Palo Alto, CA.

Doyle, L. B. (1962). Indexing and abstracting by association. *American Documentation*, 13(4):378–390.

Dumais, S. T. (1994). Latent semantic indexing (LSI) and TREC-2. In *Text Retrieval Conference (TREC-2)*, pages 105–115, Bethesda, MD.

Ekmekcioglu, F. C., Robertson, A. M., and Willett, P. (1992). Effectiveness of query expansion in ranked-output document retrieval systems. *Journal of Information Science*, 18:139–147.

Endres-Niggemeyer, B. and Neugebauer, E. (1998). Professional summarizing: No cognitive simulation without observation. *Journal of the American Society for Information Science*, 49(6):486–506.

Even, S. (1979). *Graph Algorithms.* Computer Science Press, Potomac, MD.

Everitt, B. (1980). *Cluster Analysis.* Second Edition, Heinemann Educational Books, London, England.

Fahlman, S. E. (1979). *NETL: A System for Representing and Using Real-World Knwledge.* The MIT Press, Cambridge, MA.

Finnegan, M. A. (1991). Hidden assets of controlled vocabulary when free-text searching the tulsa file of petroleum abstracts. In *Geoscience Information Society. Meeting (25th :1990 :Dallas, Tex.). Geological societies and information transfer in the electronic age.*

Foss, C. L. (1989). Tools for reading and browsing hypertext. *Information Processing and Management*, 25(4):407–418.

Fox, E. A., Nutter, J. T., Ahlswede, T., Evens, M., and Markowitz, J. (1988). Building a large thesaurus for information retrieval. In *2nd Conference on Applied Natural Language Processing, Association for Computational Linguistics*, Pages 101-108, Ballard, Bruce, Editor; Morristown, NJ: Bell Communications Research.

Frawley, W. J., Pietetsky-Shapiro, G., and Matheus, C. J. (1991). Knowledge discovery in databases: an overview. In *Knowledge Discovery in Databases*, pages 1–30, G. Piatetsky-Shapiro and W. J. Frawley, Editors, The MIT Press, Cambridge, MA.

Frieder, O. and Siegelmann, H. T. (1991). On the allocation of documents in multiprocessor information retrieval systems. In *Proceedings of the Fourteenth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, pages 230–239, Chicago, IL.

Furnas, G. W., Landauer, T. K., Gomez, L. M., and Dumais, S. T. (1983). Statistical semantics: Analysis of the potential performance of key-word information systems. *Bell System Technical Journal*, 62(6):1753–1806.

Furnas, G. W., Landauer, T. K., Gomez, L. M., and Dumais, S. T. (1987). The vocabulary problem in human-system communication. *Communications of the ACM*, 30(11):964–971.

Gallant, S. I. (1988). Connectionist expert system. *Communications of the ACM*, 31(2):152–169.

Goodman, B. A., editor (1997). *GeoRef Thesaurus*. American Geological Institute, Alexandria, VA.

Goodman, S. E., Mehrer, A. L., Lynch, K. J., and Roche, E. M. (1990). International developments in the information technologies: the Mosaic group at The University of Arizona. In *Proceedings of the 23rd Annual HICSS Conference, Decision Support and Knowledge Based Systems Track*, pages 605–611, Kailua-Kona, HI.

Gruber, T. R. (1993). A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2):199–220.

Hendry, D. G. and Harper, D. J. (1997). An informal information-seeking environment. *Journal of the American Society for Information Science*, 48(11):1036–1048.

Hersh, W. (1994). Relevance and retrieval evaluation: Perspectives from medicine. *Journal of the American Society for Information Science*, 45(3):201–206.

Hinton, G. E. and Sejnowski, T. J. (1986). Learning and relearning in Boltzmann machines. In *Parallel Distributed Processing*, pages 282–317, D. E. Rumelhart, J. L. McClelland, and the PDP Research Group, Editors, The MIT Press, Cambridge, MA.

Hopfield, J. J. (1982). Neural network and physical systems with collective computational abilities. *Proceedings of the National Academy of Science, USA*, 79(4):2554–2558.

Humphreys, B. L. and Lindberg, D. A. (1989). Building the unified medical language system. In *Proc. 13th Annual Symp. Computer Applications in Medical Care*, pages 475–480, Los Alamitos, CA. IEEE CS Press.

Institution of Electrical Engineers (1993). *INSPEC Thesaurus.* Institution of Electrical Engineers, London.

Jacquemin, C. and Tzoukermann, E. (1999). Nlp for term variant extraction: Synergy between morphology, lexicon, and syntax. In Strzalkowski, T., editor, *Natural Language Information Retrieval*, Text, Speech and Language Technology, pages 25–74. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Jain, A. K. and Dubes, R. C. (1988). *Algorithms for Clustering Data.* Prentice Hall, Englewood Cliffs, NJ.

Kao, M. L. (1995). *Cataloging and classification for library technicians.* The Haworth Press, Inc., Binghamton, NY.

Kaufman, K. A., Michalski, R. S., and Kerschberg, L. (1991). Mining for knowledge in databases: goals and general description of the INLEN system. In

Knowledge Discovery in Databases, pages 449–462, G. Piatetsky-Shapiro and W. J. Frawley, Editors, The MIT Press, Cambridge, MA.

Kirsch, S. (1998). The future of internet search: Infoseek's experiences searching the internet. *SIGIR Forum*, 32(2):3–7.

Knight, K. (1990). Connectionist ideas and algorithms. *Communications of the ACM*, 33(11):59–74.

Kohl, D. F. (1986). *Cataloging and catalogs: A handbook for library management.* ABC-Clio, Inc., Santa Barbara, CA.

Kohonen, T. (1995). *Self-Organizing Maps.* Springer-Verlag, Berlin, Heidelberg.

Kowalski, G. (1997). *Information retrieval systems: Theory and implementation.* The Kluwer International Series on Information Retrieval. Kluwer Academic Publishers, Norwell, MA.

Lancaster, F. W. (1986). *Vocabulary control for information retrieval.* Information Resources Press, Arlington, VA.

Langridge, D. W. (1989). *Subject Analysis : Principles and Procedures.* London; New York: Bowker-Saur.

Lea, R. B. (1998). Logical inferences and comprehension: How mental-logic and text processing theories need each other. In Braine, M. D. S. and O'Brien, D. P., editors, *Mental Logic*, chapter 5, pages 63–78. Lawrence Erlbaum Assocates, Mahwah, NJ.

Lea, R. B., O'Brien, D. P., Fisch, S. M., Noveck, I. A., and Braine, M. D. S. (1990). Predicting propositional logic inferences in text comprehension. *Journal of Memory and Language*, 29:361–387.

Lehmann, F. (1992). Semantic networks. In Lehmann, F., editor, *Semantic Networks in Artificial Intellgence.* Pergamon Press, Tarrytown, NY.

Lenat, D. B. and Guha, R. V. (1990). *Building large knowledge-based systems.* Addison-Wesley, Reading, MA.

Lenat, D. B., Guha, R. V., Pittman, K., Pratt, D., and Shepherd, M. (1990). Cyc: Toward programs with common sense. *Communications of the ACM*, 33(8):30–49.

Library of Congress. Subject Cataloging Division. (1998). *Library of Congress subject headings*. Library of Congress, 15th edition.

Lindberg, D. A. and Humphreys, B. L. (1990). The UMLS knowledge sources: Tools for building better user interface. In *Proceedings of the Fourteenth Annual Symposium on Computer Applications in Medical Care*, pages 121–125, Los Alamitos, CA: Institute of Electrical and Electronics Engineers.

Lindberg, D. A. B., Humphreys, B. L., and McCray, A. T. (1993). The Unified Medical Language System. *Methods of Information in Medicine*, 32:281–291.

Lippmann, R. P. (1987). An introduction to computing with neural networks. *IEEE Acoustics Speech and Signal Processing Magazine*, 4(2):4–22.

Lipton, P. (1991). *Inference to the best explanation*. Routledge, London.

Luhn, H. P. (1960). Keyword in context index for technical literature (kwic index). *American Documentation*, 11:288–295.

Lynch, K. J., Snyder, J. M., McHenry, W. K., and Vogel, D. R. (1990). The Arizona Analyst Information System: Supporting collaborative research on international technological trends. In *Proceedings of the IFIP WG8.4 Conference on Multi-user Interfaces and Applications*, pages 159–174, Crete, Greece.

Ma, W.-Y. and Manjunath, B. S. (1998). A texture thesaurus for browsing large aerial photographs. *Journal of the American Society for Information Science*, 49(7):633–648.

Marchionini, G. and Shneiderman, B. (1988). Finding facts vs. browsing knowledge in hypertext systems. *IEEE Computer*, 21(1):70–80.

Maron, M. E. and Kuhns, J. L. (1960). On relevance, probabilistic indexing and information retrieval. *Journal of the ACM*, 7(3):216–243.

McCray, A. T. and Hole, W. T. (1990). The scope and structure of the first version of the UMLS semantic network. In *Proceedings of the Fourteenth Annual Symposium on Computer Applications in Medical Care*, pages 126–130, Los Alamitos, CA: Institute of Electrical and Electronics Engineers.

McCray, A. T. and Nelson, S. J. (1995). The representation of meaning in the UMLS. *Methods of Information in Medicine*, 34:193–201.

McHenry, W. K., Lynch, K. J., and Snyder, J. M. (1990). Cross-cultural information technologies research: the case of Eastern Europe and the USSR. In *Proceedings of the Eleventh International Conference on Information Systems*, Copenhagen, Denmark.

McLachlan, G. J. (1992). *Discriminant analysis and statistical pattern recognition*. Wiley, New York.

Meghabghab, D. B. (1995). Cd-rom vs. online vs. internet: Search strategies and evaluation from the user's perspective. In Williams, E., editor, *Proceedings of the 16th National Online Meeting*, pages 295–307, Medford, NJ. Learned Information.

Michalski, R. S. and Larson, J. B. (1978). Selection of most representative training examples and incremental generation of VL1 hypotheses: the underlying methodology and the description of programs ESEL and AQ11. In *Department of Computer Science*, University of Illinois at Urbana-Champaign.

Michalski, R. S., Mzetic, I., Hong, J., and Lavrac, N. (1986). The multipurpose incremental learning system aq15. In *Proceedings AAAI-86: Fifth National Conference on Artificial Intelligence, August 11-15, 1986, Philadelphia, PA*, pages 1041–1045, Los Altos, CA. Morgan Kaufmann Publishers, Inc.

Michalski, R. S. and Stepp, R. E. (1983). Learning from observation: conceptual clustering. In *Machine Learning, An Artificial Intelligence Approach*, Pages 331-363, Michalski, R. S., Carbonell, J. G., and Mitchell, T. M., Editors, Tioga Publishing Company, Palo Alto, CA.

Miller, R. E. and Terwillegar, J. C. (1990). *Commonsense Cataloging: A Cataloger's Manual*. H.W. Wilson, New York.

Minsky, M. (1968). Introduction. In Minsky, M., editor, *Semantic Information Processing*, chapter 1. MIT Press, Cambridge, MA.

Monarch, I. and Carbonell, J. G. (1987). CoalSORT: A knowledge-based interface. *IEEE EXPERT*, pages 39–53.

Montgomery, D. D. (1976). *Design and Analysis of Experiments.* John Wiley & Sons, New York, NY.

Morrison, D. F. (1976). *Multivariate Statistical Methods.* McGraw-Hill Book Company, New York, NY.

Nelson, C. R. (1973). *Applied Time Series Analysis for Managerial Forecasting.* Holden-Day, Inc., San Francisco, CA.

Newell, A. and Simon, H. A. (1972). *Human Problem Solving.* Prentice-Hall, Englewood Cliffs, NJ.

Nunamaker, J. F., Chen, M., and Purdin, T. D. M. (1991). Systems development in information systems research. *Journal of Management Information Systems*, 7(3):89–106.

O'Brien, D. P. (1998). Logical inferences and comprehension: How mental-logic and text processing theories need each other. In Braine, M. D. S. and O'Brien, D. P., editors, *Introduction: Some Background to the Mental-logic Theory and to the Book*, chapter 5, pages 1–6. Lawrence Erlbaum Assocates, Mahwah, NJ.

Oddy, R. N. and Balakrishnan, B. (1991). PTHOMAS: an adaptive information retrieval system on the connection machine. *Information Processing and Management*, 27(4):317–335.

Parsaye, K., Chignell, M., Khoshafian, S., and Wong, H. (1989). *Intelligent Databases.* John Wiley & Sons, Inc., New York, NY.

Parsaye, K., Chignell, M., Khoshafian, S., and Wong, H. (1990). Intelligent databases. *AI Expert*, 5(3):38–47.

Peat, H. J. and Willett, P. (1991). The limitations of term co-occurrence data for query expansion in document retrieval systems. *Journal of the American Society for Information Science*, 42(5):378–383.

Plaunt, C. and Norgard, B. A. (1998). An association-based method for automatic indexing with a controlled vocabulary. *Journal of the American Society for Information Science*, 49(10):888–902.

Pool, R. (1995). Off-the-shelf chips conquer the heights of computing. *Science*, 269:1359–1361.

Quillian, M. R. (1968). Semantic memory. In *Semantic Information Processing*, M. Minsky, Editor, The MIT Press, Cambridge, MA.

Quinlan, J. R. (1983). Learning efficient classification procedures and their application to chess end games. In *Machine Learning, An Artificial Intelligence Approach*, Pages 463-482, Michalski, R. S., Carbonell, J. G., and Mitchell, T. M., Editors, Tioga Publishing Company, Palo Alto, CA.

Rada, R. and Martin, B. K. (1987). Augmenting thesauri for information systems. *ACM Trans. Office Information Systems*, 5(4):378–392.

Raghavan, V. V., Jung, G. S., and Bollmann, P. (1989). A critical investigation of recall and precision as measures of retrieval system performance. *ACM Transactions on Office and Information Systems*, 7(3):205–229.

Rao, R., Card, S. K., Jellinek, H. D., Mackinlay, J. D., and Robertson, G. G. (1992). The information grid: A framework for information retrieval and retrieval-centred applications. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 23–32, Monterey, CA. New York: ACM Press.

Rasmussen, E. (1992). Clustering algorithms. In *Information Retrieval: Data Structures and Algorithms*, W. B. Frakes and R. Baeza-Yates, Editors, Prentice Hall, Englewood Cliffs, NJ.

Rasmussen, E. M. (1991). Introduction: parallel processing and information retrieval. *Information Processing and Management*, 27(4):255–263.

Reeves, W. W. (1996). *Cognition and Complexity: The Cognitive Science of Managing Complexity.* Scarecrow Press, Inc., Lanham, MD.

Rich, E. and Knight, K. (1991). *Artificial Intelligence.* McGraw-Hill, Inc., New York, NY, second edition.

Rocchio, J. J. (1971). Relevance feedback in information retrieval. In Salton, G., editor, *The SMART Retrieval Storage and Retrieval System*, pages 313–23. Prentice Hall, Inc., Englewood Cliffs, NJ.

Roget, P. M. (1962). *Roget's International Thesaurus*. Crowell, New York, 3rd edition.

Rosch, E. (1978). Principles of categorization. In Rosch, E. and Lloyd, B. B., editors, *Cognition and Categorization*, chapter 2, pages 27–48. Lawrence Erlbaum Associates, Hillsdale, NJ.

Rouse, W. B., Thomas, B. S., and Boff, K. R. (1998). Knowledge maps for knowledge mining: Application to r&d/technology management. *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews*, 28(3):309–317.

Rowley, J. E. (1988). *Abstracting and Indexing*. London : Bingley.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning internal representations by error propagation. In *Parallel Distributed Processing*, pages 318–362, D. E. Rumelhart, J. L. McClelland, and the PDP Research Group, Editors, The MIT Press, Cambridge, MA.

Ryan, B. F., Joiner, B. L., and Ryan, T. A. (1985). *MINITAB Handbook, 2nd Edition*. PWS-KENT Publishing Company, Boston, MA.

Salton, G. (1975). *A Theory of Indexing*, volume 18 of *Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics, Philadelphia, PA.

Salton, G. (1988). *Automatic Text Processing : The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, Reading, MA.

Salton, G., Allan, J., and Buckley, C. (1994). Automatic structuring and retrieval of large text files. *Communications of the ACM*, 37(2):97–108.

Salton, G. and Buckley, C. (1988). Parallel text search methods. *Communications of the ACM*, 31(2):202–215.

Salton, G. and Buckley, C. (1990). Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41(4):288–297.

Salton, G., Wong, A., and Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.

Salton, G. and Yang, C. S. (1973). On the specification of term values in automatic indexing. *Journal of Documentation*, 29(4):351–372.

Salton, G. and Yu, C. T. (1973). On the construction of effective vocabularies for information retrieval. In *SIGPLAN/SIGIR Symposium on Programming Languages and Information Retrieval*, Gaithersburg, MD.

Schwartz, C. (1998). Web search engines. *Journal of the American Society for Information Science*, 49(11):973–982.

Shastri, L. (1991). Why semantic networks? In *Principles of Semantic Networks: Explorations in the Representation of Knowledge*, pages 109–136, J. F. Sowa, Editor, Morgan Kauffmann Publishers, Inc., San Mateo, CA.

Shneiderman, B. (1997). *Designing the User Interface: Strategies for Effective Human-computer Interaction*. Addison-Wesley, Reading, MA.

Shortliffe, E. H. (1976). *Computer-based medical consultations: MYCIN*. Elsevier, New York.

Shoval, P. (1985). Principles, procedures and rules in an expert system for information retrieval. *Information Processing & Management*, 21(6):475–487.

Sievert, M. C. and Andrews, M. J. (1991). Indexing consistency in information science abstracts. *Journal of the American Society for Information Science*, 42:1–6.

Simpson, P. K. (1990). *Artificial Neural Systems: Foundations, Paradigms, Applications, and Implementations*. McGraw-Hill Book Company, New York, NY.

Sowa, J. F. (1991a). Issues in knowledge representation. In *Principles of Semantic Networks: Explorations in the Representation of Knowledge*, pages 1–12, J. F. Sowa, Editor, Morgan Kauffmann Publishers, Inc., San Mateo, CA.

Sowa, J. F. (1991b). *Principles of Semantic Networks.* Morgan Kaufmann Publishers, Inc., San Mateo, CA.

Sparck Jones, K. (1971). *Automatic Keyword Classification for Information Retrieval.* Butterworths, London.

Stanfill, C. and Thau, R. (1991). Information retrieval on the connection machine: 1 to 8192 gigabytes. *Information Processing and Management*, 27(4):285–310.

Stiles, H. E. (1961). The association factor in information retrieval. *Journal of the Association of Computing Machinery*, 8(2):271–279.

Tague-Sutcliffe, J. (1992). Measuring the informativeness of a retrieval process. In *Proc. of the 15th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 23–36, Copenhagen, Denmark.

Tank, D. W. and Hopfield, J. J. (1987). Collective computation in neuronlike circuits. *Scientific American*, 257(6):104–114.

Thomborson, C. D. (1993). Does your workstation computation belong on a vector supercomputer? *Communications of the ACM*, 36(11):41–49.

Touretzky, D. and Hinton, G. E. (1988). A distributed connectionist production system. *Cognitive Science*, 12(3):423–466.

Tufte, E. R. (1990). *Envisioning Information.* Graphics Press, Cheshire, CT.

Tufte, E. R. (1997). *Visual explanations: Image and quantities, evidence and narrative.* Graphics Press, Cheshire, CT.

Vickery, B. and Vickery, A. (1993). Online search interface design. *The Journal of Documentation*, 49(3):103–187.

Visser, W. (1994). Organisation of design activities: Opportunistic, with hierarchical episodes. *Interacting with Computers*, 6(3):239–274.

Wactlar, H. D., Christel, M. G., Gong, Y., and Hauptmann, A. G. (1999). Lessons learned from building a terabyte digital video library. *IEEE Computer*, 32(2):66–73.

Wah, B. (1993). Report on workshop on high performance computing and communications for grand challenge applications: computer vision, speech and natural language processing, and artificial intelligence. *IEEE Transactions on Knowledge and Data Engineering*, 5(1):138–154.

Ward, J. (1963). Hierarchical grouping to optimize an objection function. *Journal of the American Statistical Association*, 58:236–244.

Winston, P. H. (1984). *Artificial Intelligence, Second Edition*. Addison-Wesley Publishing Company, Inc, Reading, MA.

Witten, I. H., McNab, R. J., Jones, S., Apperley, M., Bainbridge, D., and Cunningham, S. J. (1999). Managing complexity in a distributed digital library. *IEEE Computer*, 32(2):74–79.

Wrobel, S. (1994). *Concept Formation and Knowledge Revision*. Kluwer Acedemic Publishers, The Netherlands.

Wu, S. and Manber, U. (1992a). Agrep - a fast approximate pattern-matching tool. In *Usenix Winter 1992 Technical Conference*, pages 153–162, San Francisco.

Wu, S. and Manber, U. (1992b). Fast text searching allowing errors. *Communications of the ACM*, 35:83–91.

Wurman, R. S. (1989). *Information anxiety*. Random House, New York.

Wynar, B. S. (1985). *Introduction to cataloging and classification*. Library Science Text Series. Libraries Unlimited, Inc., Littleton, CO, seventh edition.

Yang, J. and Korfhage, R. R. (1993). Effects of query term weights modification in document retrieval: a study based on a genetic algorithm. In *Proceedings of the Second Annual Symposium on Document Analysis and Information Retrieval*, pages 271–285, Las Vegas, NV.