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:

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

$\mu_i(t)$ 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.