Generating, Integrating, and Activating Thesauri for Concept-Based Document Retrieval

Hsinchun Chen, Kevin J. Lynch, Koushik Basu, and Tobun Dorbin Ng, University of Arizona

THIS BLACKBOARD-BASED DESIGN USES A NEURAL-NET SPREADING-ACTIVATION ALGORITHM TO TRAVERSE MULTIPLE THESAURI. GUIDED BY HEURISTICS, THE ALGORITHM ACTIVATES RELATED TERMS IN THE THESAURI AND CONVERGES ON THE MOST PERTINENT CONCEPTS.

THE UNSTRUCTURED NATURE and volume of textual information make the difficult task of information retrieval even more problematic. Document management systems provide little support for suggesting appropriate search terms or helping users articulate their queries, leaving users to search the document space themselves, using the system's basic pattern-matching capability.

Researchers and practitioners have proposed various AI methods to help users choose search terms and articulate queries. (See the sidebar on related work, pp. 26-27.) A widely accepted approach is to incorporate thesaurus-like components that represent important concepts in the subject area and their semantic relationships. Thesauri can be created from existing sources such as *Roget's Thesaurus*, developed by users, or generated automatically by applying statistics-based cluster analysis techniques to stored documents.

When a document management system incorporates a thesaurus, users can ask it to find synonyms or related concepts. The process of continuously tracing cross-referencing relationships in the thesaurus helps increase the search recall (a measure of how well a user retrieves all relevant documents). The thesaurus can be viewed as a *concept space*, in which users browse and relate various domain-specific concepts and significantly augment their mental perception and knowledge about specific domains.

Still, both the development and use of thesauri in document management systems are problematic. First, existing thesauri often represent a general subject area, so they usually need significant enhancement to be tailored to a specific application. User-defined or system-generated thesauri represent subject-area knowledge better than existing thesauri, but the concepts represented may be too specific (that is, without proper context) to be useful to novice users. For systems with multiple thesauri, there is an obvious problem of how to use and consolidate concepts from different knowledge sources.

Also, the thesaurus lookup and term switching (traversing a single cross-reference link) supported by most document management systems can lead to two design problems that resemble hypertext's browsing difficulties: the "embedded digression problem," in which a large thesaurus confuses and disorients users, and the "art museum phenomenon," in which users spend a great deal of time learning nothing specific.[1]

Our document management system uses a neural-net spreading-activation algorithm that lets users traverse multiple thesauri. Guided by heuristics, the algorithm activates related terms in the thesauri and converges on the most pertinent concepts.

0885/9000/93/0400-0025 $3.00 © 1993 IEEE

Related projects

Many researchers in information science and computer science have tried in recent years to capture an expert's domain knowledge for information retrieval. One example is CoalSort,[1] a knowledge-based interface that facilitates the use of bibliographic databases in coal technology. A semantic network representing an expert's domain knowledge embodies the system's intelligence. The Grant expert system[2] finds sources of funding for research proposals. Its search method—constrained spreading activation in a semantic network—makes inferences about the user's goals and thus finds information that the user did not explicitly request but that is likely to be useful. The expert system developed by Shoval[3] suggests search terms. It is composed of a knowledge base and rules. The knowledge base is represented as a semantic network in which the nodes are words, concepts, or phrases. Links express the semantic relationships between nodes. The rules, or procedures, operate on the knowledge base and are analogous to the decision rules or work patterns of the information specialist. The Coder system[4] uses a thesaurus generated from the *Handbook of Artificial Intelligence* and *Collin's Dictionary*. The Intelligent Intermediary for Information Retrieval,[5] or I^{3}R, consists of a group of "experts" that communicate via a common data structure, called the blackboard. It includes a user model builder, a query model builder, a thesaurus expert, a search expert (for suggesting statistics-based search strategies), a browser expert, and an explainer. Chen and Dhar[6] incorporated a portion of the Library of Congress Subject Headings into the design of an intelligent retrieval system that uses a branch-and-bound spreading-activation algorithm to help users articulate queries.

The National Library of Medicine's thesaurus projects are probably the largest-scale effort that use the knowledge in existing thesauri. In one project, Rada and Martin[7] experimented with automatically adding concepts to Mesh (Medical Subject Headings) using two thesauri, the Current Medical Information and Terminology and the Systematized Nomenclature of Medicine. The Unified Medical Language System project is a long-term effort 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 machine-readable sources.[8-9] The UMLS includes a metathesaurus, a semantic network, and an information sources map. The metathesaurus contains information about biomedical concepts and their representation in about 10 different vocabularies and thesauri. The semantic network contains information about the types of terms (such as disease, virus, and so on) 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. Most of the knowledge bases used in these systems were either generated manually from the domain experts via the knowledge acquisition process or derived from existing thesauri (which were first created manually by some indexing or subject experts).

Complementary to manual knowledge base creation is the automatic thesaurus-generation approach, which can be extremely useful in capturing the specific domain knowledge in textual databases. Virtually all techniques for automatic thesaurus generation are based on the statistical co-occurrence of word types in text.[10-12] Similarity coefficients are often obtained between pairs of distinct terms based on coincidences in term assignments to the documents of the collection. For example, a cosine computation can be used to generate normalized term similarities between 0 and 1. When pairwise similarities are obtained between all term pairs, an automatic term-classification process such as single-link or complete-link classification can group into common classes all terms with sufficiently large pairwise similarities. The terms in these classes can replace the initial search terms and be used to increase search recall.

Thesauri in commercial full-text retrieval systems

To understand the role and functionalities of thesauri in commercial full-text retrieval systems, we examined several major software packages on the market:

- Topic from Verity (Mountain View, California);
- Basis/Plus from Information Dimensions (Dublin, Ohio);
- Fulcrum Ful/Text from Fulcrum Technologies (Ottawa, Ontario, Canada);
- Elexir from Third Eye Software (San Jose, California); and
- Savvy/TRS from Excalibur Technologies (McLean, Virginia).

Each has the structural capability to incorporate a thesaurus; that is, they each have some type of thesaurus, either manually built or included from an outside source. They can also incorporate thesaurus terms during the retrieval process.

Topic is different from most full-text retrieval systems in that it performs concept-based information retrieval. Queries are created using a tree-structured hierarchy of topics, akin to a small user-defined thesaurus. The software captures and stores in topic trees the expertise of individuals knowledgeable about any given subject. Users define the relationships between subtopics, words, and phrases; express the importance

A blackboard approach for knowledge-base integration

To integrate the knowledge in various thesauri and facilitate their most effective and "intelligent" use, we adopted a blackboard architecture. This approach has proven useful in organizing and coordinating complex knowledge sources and in assisting joint, opportunistic problem solving. It has been applied successfully in the design of various knowledge-based systems, including Hearsay-II,[2] I^{3}R,[3] and Coder.[4] A typical blackboard architecture consists of three components: a blackboard, knowledge sources, and control modules.[2]

The blackboard. The data and the partial results involved in the problem-solving process are kept in a global working area, called the blackboard. In information retrieval applications, users are interested in documents and search terms (such as keywords, authors, organizations, and journal names). During a search, the user's own subject area knowledge, intermediate search results, thesaurus terms, and other incidental cues displayed on the screen may trigger appropriate search terms and help the user find relevant documents. Our blackboard has two working areas that act as "notepads," where users can examine the status of their search results, refine or articulate their information needs, or continue the search using the terms or documents on the blackboard.

Two types of data are posted in separate areas of the blackboard. At the first level, the user generates or selects search terms that represent the concepts in the queries. These terms are posted on the concept blackboard, line by line. During a retrieval session, these concepts may evolve due to the user articulating his or her needs, the system suggesting specific concepts, or other incidental clues triggering new ideas in the user. The user can erase terms, or mark terms to browse or activate the various thesauri.

of each piece of evidence in the tree to the overall concept; and establish rules for combining the weighted evidence in the tree. Topic can also include user-specified synonyms as well as external thesauri such as *Roget's*. Users can browse the thesaurus and include related terms in their queries. Verity said recently that it will soon offer domain-specific knowledge bases in finance and accounting.

Basis/Plus is an application development tool with a sophisticated query language, refined indexing that relies on the B-tree method, a screen manager, and security with various protection levels. The thesaurus module in Basis/Plus uses controlled vocabulary and performs term switching (from common search terms to preferred terms), and supports the 13 ANSI standard relationships (for example, broader and narrower terms). To ensure thorough retrieval, users can browse the thesaurus to find the relationships associated with specific terms. The thesaurus also supports interactive dialogue in several European languages.

Fulcrum Ful/Text is a full-text indexing and retrieval software suited to applications involving both structured and unstructured text. It contains a library of over 100 callable routines that can be used by application developers to create their own document management environment.

Elexir also provides a library of indexing, retrieval, format conversion, and interface routines that can be customized for users' applications. Both Fulcrum Ful/Text and Elexir can incorporate an external thesaurus such as *Roget's* as a search aid. These systems can search their thesaurus and allow the user to draw terms from it.

Savvy was originally developed as a library of callable C routines. This library of neural-network programs was repackaged for sale as various systems, including Savvy/TRS. With the optional thesaurus module, Savvy/TRS lets users search for synonyms through a thesaurus file that they can create or obtain from the public domain. These content-based searches use one or more synonymous or associated phrases at the same time.

Most commercial full-text retrieval systems provide some thesaurus modules, which are either user-defined or extracted from existing sources and generally deemed useful and necessary. Except for Topic's domain-specific knowledge bases, most thesauri that are generated from existing sources are not domain-specific. None of the commercial products adopts a system-generated thesaurus, and there are no effective ways other than simple browsing and term replacement to help users traverse these thesauri.

References

1. I. Monarch and J.G. Carbonell, "CoalSort: A Knowledge-Based Interface," *IEEE Expert*, Vol. 2, No. 1, Spring 1987, pp. 39-53.
2. P.R. Cohen and R. Kjeldsen, "Information Retrieval by Constrained Spreading Activation in Semantic Networks," *Information Processing and Management*, Vol. 23, No. 4, 1987, pp. 255-268.
3. P. Shoval, "Principles, Procedures, and Rules in an Expert System for Information Retrieval," *Information Processing and Management*, Vol. 21, No. 6, 1985, pp. 475-487.
4. E.A. Fox, "Development of the Coder System: A Testbed for Artificial Intelligence Methods in Information Retrieval," *Information Processing and Management*, Vol. 23, No. 4, 1987, 341-366.
5. W.B. Croft and R.H. Thompson, "I^3R: A New Approach to the Design of Document Retrieval Systems," *J. Am. Soc. for Information Science*, Vol. 38, No. 6, 1987, pp. 389-404.
6. H. Chen and V. Dhar, "Cognitive Process as a Basis for Intelligent Retrieval Systems Design," *Information Processing and Management*, Vol. 27, No. 5, 1991, pp. 405-432.
7. R. Rada and B.K. Martin, "Augmenting Thesauri for Information Systems," *ACM Trans. Office Information Systems*, Vol. 5, No. 4, Oct. 1987, pp. 378-392.
8. B.L. Humphreys and D.A. Lindberg, "Building the Unified Medical Language System," *Proc. 13th Annual Symp. Computer Applications in Medical Care*, IEEE CS Press, Los Alamitos, Calif., 1989, pp. 475-480.
9. A.T. McCray and W.T. Hole, "The Scope and Structure of the First Version of the UMLS Semantic Network," *Proc. 14th Annual Symp. Computer Applications in Medical Care*, IEEE CS Press, Los Alamitos, Calif., 1990, pp. 126-130.
10. C.J. Crouch, "An Approach to the Automatic Construction of Global Thesauri," *Information Processing and Management*, Vol. 26, No. 5, 1990, pp. 629-640.
11. H. Chen and K.J. Lynch, "Automatic Construction of Networks of Concepts Characterizing Document Databases," *IEEE Trans. Systems, Man and Cybernetics*, Vol. 22, No. 5, Sept./Oct. 1992, pp. 885-902.
12. G. Salton, *Automatic Text Processing*, Addison-Wesley, Reading, Mass., 1989.

At the second level, documents derived during the search are recorded on the document blackboard. Each document is summarized by an identifier followed by the first line of text. During information retrieval, users scan and select relevant documents, which are then recorded and can be evaluated, updated, or erased at any time during a search. As users' queries become more focused and well articulated after extensive iterations, documents posted on the blackboard will have high recall and precision.

Knowledge sources. In a blackboard architecture, the knowledge needed to accomplish a task is partitioned into different knowledge sources, each acting as a small, specific expert (or demon) ready to contribute to completing the task. In our problem domain, thesauri are viewed as individual knowledge sources that help users augment their concept space and improve search recall.

To investigate multiple-thesaurus issues, we incorporated into our prototype four thesauri generated by distinct methods from unique sources: an algorithmically generated knowledge base on Russian computing, a user-defined folder hierarchy, and two existing computing-related thesauri from public sources. The thesauri support information retrieval in an operational Russian-computing database that occupies more than 200 Mbytes and contains over 40,000 documents (articles, book chapters, product brochures, and so on). Each thesaurus has a unique structure and vocabulary and can be activated individually or with others to aid on-line concept-based information retrieval.

The Mosaic knowledge base. Mosaic KB was automatically created by cluster analysis of the documents in a database that was developed by the University of Arizona's Mosaic research group. It is our most domain-specific thesaurus, containing

knowledge most relevant to the database's subject area, Russian computing. The database, created in Ingres, was designed to support research on computing in the former East-bloc countries. Researchers from different parts of the world access this database for information retrieval and intelligence analysis. The Mosaic researchers have maintained and used the database for the past nine years in areas such as East-bloc computing evaluation, industry policy analysis, technology assessment, and export control recommendations on US information products. Information was entered through a template-driven process; it was often very unstructured, ranging from articles, book chapters, and technical reports to business cards, foreign-trip reports, and electronic-mail messages.

For each document stored in the database, descriptors such as keywords, persons, organizations, countries, and folders indicate the document's context and content. Documents of similar content, collected over time from different sources, often contain similar descriptors. The co-occurrence of descriptors in documents can reveal relationships between important topics (projects, computers, policy, and so on), crucial persons, relevant organizations, and countries in East-bloc computing.

To create Mosaic KB algorithmically, we adopted a few cluster analysis algorithms, including the cosine similarity function. The resulting knowledge base contains about 20,000 concepts (nodes) and 280,000 weighted relationships (links). Our algorithms identified five types of semantic objects in the Mosaic database and their corresponding links. Keyword objects and related-keyword links (also called related-term or RT links) describe related topics, machines, projects, and so on (for example, "technology transfer," "MVS 810"). Folder objects and related-folder (RF) links represent related user-created folders (for example, "softlaw.dat" for the Russian software protection law folder). Person objects and related-person (RP) links indicate the people related to a document (for example, "Y. Andropov"). Organization objects and related-organization links (RO links) indicate institutions related to a document (such as "Academy of Science in Kiev") and country objects and related-country (RC) links indicate related countries such as "USSR" or "Poland."

We evaluated this knowledge base in a cognitive experiment.[5] Mosaic KB outperformed four experts in recalling relevant concepts in East-bloc computing. Using the knowledge base as a concept articulation aid, the experts improved their recall significantly, but their precision worsened. Among the four thesauri developed, the Mosaic KB has proved to be the most comprehensive and domain-relevant knowledge source.

TO CREATE MOSAIC KB ALGORITHMICALLY, WE ADOPTED A FEW CLUSTER ANALYSIS ALGORITHMS. THE RESULTING KNOWLEDGE BASE CONTAINS ABOUT 20,000 CONCEPTS (NODES) AND 280,000 WEIGHTED RELATIONSHIPS (LINKS).

The Mosaic tree. The second knowledge source we included was the Mosaic directory tree, manually created by the Mosaic analysts over the past decade. The tree represents the complete folder hierarchy of research topics of special interest to the analysts. It contains nodes at different levels, indicating the topics of interest to a particular Mosaic researcher, and the leaf nodes point to specific folders.

The hierarchy consists of four levels of specificity, with the root node indicating all the countries in the former Council for Mutual Economic Assistance (established in 1949 to promote trade and economic cooperation among the USSR and its allies). The hierarchy indicates the narrower-term and broader-term relationships between nodes of various levels. Currently the tree contains about 680 leaf nodes and 10,100 folders (each leaf node associates with multiple folders). There are term and folder object types, and narrower-term, broader-term, and related-folder link types.

ACM Computing Review Classification System. The third thesaurus represents the general computing categories used by the ACM to classify computing literature. The ACM CRCS is based on a hierarchical structure with four levels of specificity. Terms fan out level by level. Although its classification structure is simpler and its subjects are less relevant to East-bloc computing than those in the Mosaic tree or Mosaic KB, it nicely represents general computing terms and their relationships.

We identified two types of terms from the ACM CRCS. The first deals with specific and unambiguous topics such as "artificial intelligence" and "machine learning," similar to Library of Congress subject headings. The second type indicates general computing-related categories that can be appended to any term in the ACM thesaurus: for example, "verification," "documentation," or "testing." We identified 18 general categories and 1,141 specific terms from this thesaurus.

We also identified five types of relationships:

- BT and NT indicate broader and narrower hierarchical relationships between specific terms, respectively;
- RT indicates associative relationships, shown in the parentheses following some terms; and
- IsA and Inst indicate is-a and instance-of 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 special cases of the broader-term and narrower-term relationships, respectively. We identified 2,922 relationships from the ACM CRCS.

Library of Congress Subject Headings. The LCSH represents general computing terms selected by the Library of Congress for classifying computing-related books. The LCSH is network-based and cross-referenced, and its terms indicate topics. Five types of relationships exist between terms:

- RT indicates an associative relationship,
- BT/NT indicate hierarchical relationships, and

```
Mosaic KB object frame:
 (Object:
    Object type: (term, person, folder,
organization, country)
    RT: (list of related terms)
    RP: (list of related persons)
    RF: (list of related folders)
    RO: (list of related organizations)
    RC: (list of related countries)
 )
Mosaic Tree object frame:
 (Object:
    Object type: (term, folder)
    NT: (list of narrower terms)
    BT: (list of broader terms)
    RF: (list of related folders)
 )
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/categories)
    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 1. Frame-based representations for the knowledge sources.

```
9-APR-1992        VIEW Main Menu        Database: MOSAIC

    (C)  Search on Country information
    (D)  Search on Directory information
    (F)  Search on File information
    (K)  Search on Keyword information
    (L)  Search on Line (full text)
    (M)  Merge files of textids
    (N)  Search on Name (person) information
    (O)  Search on OrgId (organization) information
    (R)  Search on RefId (reference) information
    (T)  Search on TextId information

(V)  View/manipulate defaults for And, Number, Select, Since, Sort, Thesaurus

(W)  View/manipulate defaults for /Nametype

Enter letter or combinations (? = Help, X = exit) ==>
```

Figure 2. Menu-driven retrieval display for database searches (Display 1).

- Use/UF (use or used for) indicates a synonymous relationship.

Our LCSH subset contains 1,142 terms and 8,141 relationships. It is more general than the other three knowledge sources in the computing domain. However, its broader scope of coverage may help novice users articulate their general, fuzzy queries.

Knowledge source structure. The frame-based representations of the four knowledge sources are stored as Ingres relations (see Figure 1). For storage reasons, we assigned a unique integer identifier to each thesaurus object. We then stored each object, its neighbors, and their relationships in a separate relation. For Mosaic KB, which includes weighted relationships between terms, the weights are also stored in the relations. Ingres provides a structured and convenient way of storing a vast amount of information in the various thesauri.

Since Mosaic KB is extremely domain-specific, it serves as the connecting structure for the other three knowledge sources. Nodes represented in the LCSH, the ACM CRCS, and the Mosaic tree overlap in some way with the descriptors in the Mosaic KB. The Mosaic tree and Mosaic KB are more directly relevant to East-bloc computing, while the LCSH and the ACM CRCS provide contextual structures for understanding general computing concepts.

Thesaurus browsing

Our system provides two control modes, a browsing module and an activation module, that determine the sequence of operations, that is, which knowledge source to use, when, and how. With the browsing module, users have full control over which knowledge sources to browse and what terms to select.

Forming queries. The system provides two methods for specifying queries: Users can state their queries at the VAX/VMS command level, or retrieve documents through the menu-driven retrieval module. When using the "View Main Menu" (see Figure 2), users can select combinations of search types. For instance, if the user enters "KO" at the prompt, the system asks the user to enter a keyword and an organization to initiate the search. If the user then enters "database" and "IBM," the equivalent command line query would be

```
$ View / Thesaurus / Keyword =Database
 / OrgId = IBM
```

In either case, search terms are elicited and recorded in the blackboard's concept space area.

Matching terms and refining queries. The system then helps the user match terms and refine queries. Research has shown that query terms often differ from index terms (the terms-matching problem). Bates argues that for a successful match, the user must generate as much "variety" (in the cybernetic sense) in the search as is produced by the indexers in their indexing.[6] In terms of query refinement, users often do not have "queries," but what Belkin calls an "anomalous state of knowledge."[7] Users often expect to refine this anomalous state into a query through an interactive process.

Users' initial search terms are taken as the triggers to identify other semantically relevant indexes from the various knowledge sources. The system consults the thesauri, activates and ranks relevant indexes, and then lists the thesauri that match the search terms. Users are free to browse any of the matched thesauri, and their selected terms are maintained in the concept space.

Terms suggested by the system also serve as clues to help users articulate their needs. Interaction continues iteratively as the user selects and marks relevant terms, activates thesaurus terms, makes more selections, activates more thesaurus terms, and so on. During this human-system interaction cycle, the thesauri can become a concept exploration or convergence aid, alleviating the cognitive demand on users to refine their "anomalous state of knowledge."

Each knowledge source is explored in turn. Besides being able to delete and refine terms, users can select terms from

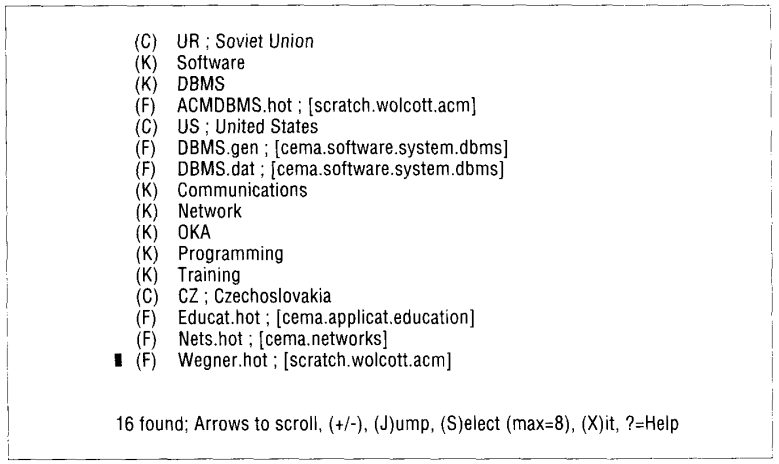


Figure 3. The Mosaic KB terms related to "database management" (Display 2).

each knowledge source to move to the blackboard's concept space. They may go out to the text at any time and retrieve a ranked list of documents based on the terms in the concept space at that time. Figure 3 shows the top-ranked Mosaic KB terms suggested by the system with an initial query request for "database management." These terms are ranked in decreasing order of relatedness, and their object types are also displayed: (F) stands for folder, (K) for keywords, (O) for organization, (N) for person name, and (C) for country. Figures 4 and 5 show the related ACM CRCS and LCSH terms, respectively, for "database management."

The lists in Figures 3, 4, and 5 appear on consecutive screens, giving users the opportunity to manipulate the Mosaic KB, ACM CRCS, and LCSH terms, in that order. After terms are selected from each screen, the concept space is displayed once again. From this point, users can delete terms, iterate inside the thesaurus once again, or go directly to a ranked document set based on the terms in the concept space.

Retrieving, ranking, and selecting documents. When users feel comfortable with their articulated queries, they can activate the system's document retrieval module, which uses the marked terms in the concept space to search the complete database. Each document retrieved is assigned a score by computing the number of matched indexes in the document. The system presents a summary table showing the number of documents matched with the different numbers of indexes. For example, for a request of five query terms, the system may find 34 documents that have all five terms as indexes, 234 documents with four matched terms, 550 documents with three matched terms, and so on.

Documents are presented in decreasing order of relatedness in a summary format: a document identifier plus the first line of text. Users can browse the complete contents of these retrieved documents or can jump from one document to another. A sample retrieved document is shown in Figure 6. The selections at the bottom of the menu indicate the options users have for retrieving document-related information (such as index and country) or performing document-specific operations (such as update or output). During document browsing, users can mark the documents they deem relevant, and these documents will be posted in the blackboard document space.

As a future extension of our current implementation, users will be able to mark documents posted on the document space and request that the system perform a "concept-based" relevance feedback, including

- identifying the indexes in the selected documents,
- automatically activating relevant concepts in the various thesauri,
- performing a document search using the initial document indexes and the activated thesaurus terms, and
- ranking the retrieved documents in decreasing order of relatedness.

In contrast to the conventional relevance feedback method, which uses the initial document indexes to perform the document search, a "concept-based" relevance feedback method uses the various thesauri to identify other semantically relevant, but syntactically different terms.

Implementation. We have incorporated the thesaurus-browsing module into the Mosaic search environment. Some researchers have indicated that the thesauri, especially the domain-specific Mosaic KB, were excellent tools for assisting query articulation. Mosaic KB helped reveal the explicit semantic relationships between objects of interest. In particular, it provided semantic interpretation (for example, related keywords, persons, and organizations) for previously obscure folders (created by researchers in the past). Some researchers even suggested using the system's knowledge about the folders to filter incoming documents and to make automatic, semantics-based folder assignments. Junior researchers also used our knowledge sources as learning aids, exploring and traversing the knowledge network to become familiar with topics of interest.

Thesaurus activation

The rich semantics and cross-references provided in the various thesauri allow users to easily enter, explore, and navigate in a network of knowledge. However, to help users focus their attention and browse effectively in a huge network/hierarchy of concepts and to avoid the classical problems described earlier, we need active and intelligent means for assisting traversal. With the activation module, users can release control and let the system's underlying neural-network algorithm traverse the knowledge sources, automatically activate associated terms in the thesauri, and make suggestions. This process is generally transparent to users.

Spreading activation, a memory association mechanism originating from human-memory research, has been used successfully in various applications. For example, semantic-net-based retrieval systems such as Grant have adopted heuristics-guided

spreading-activation algorithms.[8-9] The concept has also been applied in neural-net-based retrieval systems, such as the Adaptive Information Retrieval system.[10] Even though both representations are directed graphs, semantic nets typically represent the logical, labeled relationships between nodes, while neural nets represent weighted, probabilistic links between nodes. However, this distinction has become blurred by the development of more complex, hybrid systems that incorporate both semantic and neural nets.[11-12] Our system, which includes thesauri based on labeled links (LCSH, ACM CRCS, and the Mosaic tree) and weighted links (Mosaic KB), is also a hybrid system.

The Hopfield net algorithm is a classical method of inferencing in a weighted network.[13] The Hopfield net can be used as associated memory, where unknown input patterns (for example, fuzzy queries) can be classified and disambiguated based on the knowledge embedded in the network. By assigning normalized weights to all the labeled links based on the weights associated with Mosaic KB, we can create four networks of labeled, probabilistic links that can be activated by a Hopfield-net-like algorithm.

Our weighted network of knowledge sources can be viewed as interconnections of neurons and synapses in a Hopfield net, where neurons represent concepts (keywords, people, organizations), and synapses represent weighted links between concept pairs. The four thesauri can be considered a trained Hopfield net that enables users' initial query terms (not yet articulated and refined) to be perceived as "noisy" input. The noisy input pattern needs to be disambiguated based on the network's knowledge about the domain. By applying the Hopfield net algorithm iteratively—activating the neurons and synapses and using the weights and an input/output transformation function—output (relevant terms for the fuzzy queries) will converge and suggestions can be made concerning the terms that are most relevant to the user's queries. (A good overview of Hopfield and other neural-net algorithms can be found elsewhere.[14-15])

Our activation implementation incorporates the basic Hopfield net iteration and convergence ideas. However, we also modified it significantly to deal with four different networks and their structures. Our

```
(K)  Information systems
(K)  Database management—general
(K)  Logical design
(K)  Physical design
(K)  Database management—languages
(K)  Database management—systems
(K)  Heterogeneous databases
(K)  Database machines
(K)  Database administration
(K)  Database applications
(K)  Database management—miscellaneous
(K)  Files

12 found; Arrows to scroll, (+/-), (J)ump, (S)elect (max=7), (X)it, ?=Help
```

Figure 4. The ACM CRCS terms related to "database management" (Display 3).

```
(K)  Electronic data processing
(K)  Electronic digital computers—programming
(K)  Information storage and retrieval systems
(K)  Burroughs ISAM (computer system)
(K)  Database management—computer programs
(K)  Database management—directories
(K)  Data compression (computer science)
(K)  Data dictionaries
(K)  David (information retrieval system)
(K)  DBS/R (computer system)
(K)  Delta (computer programs)
(K)  File organization (computer science)
(K)  IBM Dbase 2 (computer system)
(K)  IDMS (computer system)
(K)  IDMS/R (computer system)
(K)  Image/3000 (computer system)
(K)  IMS (DL/I) (computer system)
(K)  IMS/VS (computer system)
(K)  Ingres (computer system)
(K)  Input design, computer

33 found; Arrows to scroll, (+/-), (J)ump, (S)elect (max=7), (X)it, ?=Help
```

Figure 5. The LCSH terms related to "database management" (Display 4).

```
KEYWORD/ORG          HANDICAP*/IBM          Total: 7        Viewing #: 3
Ref Id: Kiri90   Pg: 7      Adder: WKM      Seccode: P      Textid: 37984

This article is about a gift made by IBM (International Business Machines) to equip a
school for the deaf and one for the blind, plus two other high schools, with PCs. They will
be used to replace old equipment, as a demonstration center, and obviously to teach the
deaf to speak, etc. After pointing out that for social expenditures such as this IBM is not
taxed, the article states that IBM will equip another 20 rooms for high schools and another
20 for the deaf and blind, over the next five years.
handicapped

(C)ountry (D)elete (E)xtract (F)ile (G)em (I)ndex (J)ump (K)eyword
(L)ocation (M)ove #=Organization (O)utput $=Pause (P)erson (R)efid
(S)earch (T)ransfer (U)pdate (V)irtual file (X)it Ctrl_W=Refresh screen
(<,>,+,-,?)
```

Figure 6. A retrieved document.

```
Let knowledge sources weights be:
  KW(Mosaic KB) : KW(Mosaic tree) : KW(ACM) : KW(LCSH) = a : b : c : d;
                                        {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 Mosaic KB;
                                        {Computed from Ingres relations}
Assign weights to links in Mosaic tree:    {b/a: relative weight for M-tree}
  LW(NT) := LW(RF) := b/a * y/x * ART;      {NT and RF are specific links}
  LW(BT) := b/a * (z/x * ART);
Assign weights to links in ACM CRCS:       {c/a: relative weight for ACM}
  LW(RT) := c/a * ART;
  LW(NT) := LW(Inst) := c/a * (ART * y/x);  {Inst is special case of NT}
  LW(BT) := LW(IsA)  := c/a * (ART * z/x);  {IsA is special case of BT}
Assign weights to links in LCSH:           {d/a: relative weight for LCSH}
  LW(RT) := d/a * ART;
  LW(NT) := d/a * (ART * y/x);
  LW(BT) := d/a * (ART * z/x);
  LW(Use/UF) := 1;                         {Weight for synonymous link is 1}
```

Figure 7. Connection weight assignments for the knowledge sources.

```
Knowledge sources weights:
  KW(Mosaic KB) : KW(Mosaic tree) : KW(ACM) : KW(LCSH) = 1 : 1 : 1 : 1;
                                        {Default setting}
Link weights:
  LW(RT) : LW(NT) : LW(BT) = 2 : 3 : 1;     { NT > RT > BT, in this example}
ART :=  .203;
                                        {Computed from Ingres relations}
Weights of links in Mosaic tree:
  LW(NT) := LW(RF) := 1/1 * 3/2 * .203 = .3045;
  LW(BT) := 1/1 * (1/2 * .203) = .1015;
Weights of links in ACM CRCS:
  LW(RT) := 1/1 * .203 = .203;
  LW(NT) := LW(Inst) := 1/1 * (.203 * 3/2) = .3045;
  LW(BT) := LW(IsA)  := 1/1 * (.203 * 1/2) = .1015;
Weights of links in LCSH:
  LW(RT) := 1/1 * .203 = .203;
  LW(NT) := 1/1 * (.203 * 3/2) = .3045;
  LW(BT) := 1/1 * (.203 * 1/2) = .1015;
  LW(Use/UF) := 1;
```

Figure 8. Link weights for the knowledge sources.

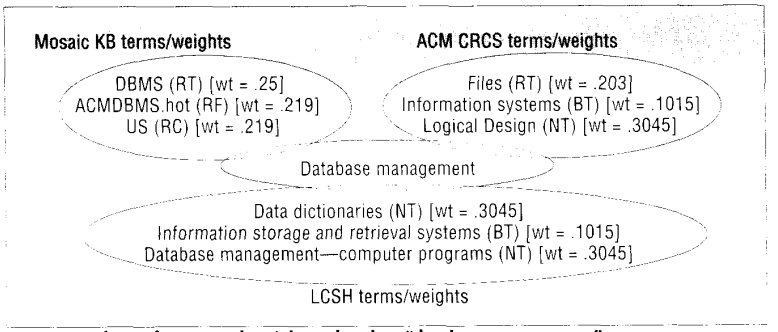


Figure 9. Subset of terms and weights related to "database management".

algorithm activates the four knowledge sources and different link types based on criteria supplied by the user. It then applies the system's iteration and convergence mechanisms. During the iteration process, the system constantly accesses the thesaurus relations in Ingres, identifies related concepts, and performs computations based on the weighted links. Our thesaurus activation procedure has four steps.

Step 1: Eliciting activation criteria. The user must supply two activation criteria: weights assigned to individual knowledge sources, and weights assigned to different types of links. A scale of 0 to 10 for each knowledge source indicates the user's preferred sources: 0 indicates the lowest preference (that is, the source is considered irrelevant), and 10 indicates the highest preference. The ratings are used to determine the relative weights associated with different knowledge sources. Another 0-to-10 scale is used to elicit the user's preferences among three types of links: broader, narrower, and related terms. For example, assigning a higher rating to narrower-term links than to broader-term ones indicates the user's intention to traverse toward more specific concepts. The Hopfield net algorithm uses these ratings to determine the activation direction. By allowing users to indicate their preferences of knowledge sources and link types, our activation algorithm can traverse the knowledge sources more effectively and intelligently.

Step 2: Assigning connection weights. The connection weight t_{ij} connects node i to node j in the knowledge sources and ranges from 0 and 1. For Mosaic KB, the weights between nodes have already been generated and stored in the Ingres relations. Because all four knowledge sources have some form of related-term relationship, we can use the weighted RT links in Mosaic KB (a neural net) as the basis for assigning weights to the RT links in the other three knowledge sources (semantic nets). This process of adopting neural-net weights in semantic nets is important in creating a hybrid system with weighted, labeled networks of concepts. The activation criteria obtained from the prior step can then be used to modify the assigned weights to reflect the user's search criteria. Figure 7 shows some connection weight assignments for the different knowledge sources.

Figure 8 shows sample link weights for the knowledge sources. This example uses the default weighting scheme for knowledge sources, where each knowledge source has an equal weighting. The link weights are set so that narrower terms are weighted more heavily than related terms, and related terms are weighted more heavily than broader terms. Figure 9 shows a subset of the terms related to "database management" and the term weights using this weighting system for knowledge sources and links.

Step 3: Initializing with search terms. In the formula

$$\mu_i(0) := x_i,\ 1 \le i \le N$$

$\mu_i(t)$ is the output of node i at time t; N is the total number of nodes in the four knowledge sources; and x_i, a number between 0 and 1, indicates the probability of the presence of a search term. At time 0, the beginning of iteration, nodes that match search terms are assigned a probability of 1.

Step 4: Iterating until convergence. Having the initial inputs and the weights, the algorithm activates neighboring terms, combines weighted links, performs the sigmoid transformation function

$$\mu_j(t+1) = \int_s \left[\sum_{i=1}^{N} t_{ij}\mu_i(t) \right], 1 \le j \le N$$

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.

This iteration process is the most time-consuming bottleneck. To obtain a node's neighboring terms, our algorithm needs to access an Ingres relation, which is a slow, I/O-bound procedure. When activation begins to fan out, more Ingres relations must be accessed. After repeated testing, we have selected some thresholds in our transformation function that help activate a reasonable number of neighboring terms.

Current status

We developed the current prototype system in C, with the support of an underlying Ingres database management system. Some Ingres access routines were written in Fortran. All four knowledge sources are incorporated into our system as thesaurus components. The thesaurus-browsing module is operational in the Mosaic environment, and we are now fine-tuning the thesaurus activation module.

We evaluated the prototype implementation of the thesaurus activation module to consider the feasibility and limitations of our approach. We tested several queries with one, two, three, four, and five terms each. For each test case, we observed the number of iterations, the terms suggested during each iteration, the sources of the activated terms, and the activation time. Since the thesauri are large, dynamic loading of the entire network exceeded the memory capacity of the host machine. We decided to construct the neural network "on the fly." Since the neighboring terms and associated weights for any given node are all stored in the underlying Ingres relational tables, the network is instantiated and activated at runtime, growing in size with each iteration as more and more terms are activated.

Not surprisingly, the Ingres table access was the bottleneck for our activation process. The I/O time required for the ith iteration is approximately

$$P \times (N_{KB} + N_{tree} + N_{ACM} + N_{LCSH})^i \times A_i$$

if the time taken for the disk access is A_i, the average fan-outs of a node in the four knowledge sources are N_{KB}, N_{tree}, N_{ACM}, and N_{LCSH}, and there are P search terms.

Setting thresholds so that each neuron is activated only when its output exceeds a certain point helps reduce this exponential search space significantly and ensure fast convergence. In our preliminary experiment, the number of iterations that were activated in the knowledge sources rarely exceeded four. Increasing the threshold led to faster convergence but decreased activation levels because of the greater damping factor. Our threshold level was determined empirically, which led to reasonable retrieval times and activation levels. Our current prototype performed a complete thesaurus activation process on the four thesauri for one-term queries in less than 20 seconds. For four- and five-term queries, the access time increased proportionally to about one to two minutes; for cases where search terms were related, it converged sooner. We believe we can improve this access time by using more efficient search routines and a more powerful computer.

For Russian-computing-specific queries (for example, "BESM-6," a Soviet high-performance computer), most terms were suggested by Mosaic KB and the Mosaic tree, and were often a few links away from the starting terms. The LCSH and the ACM CRCS were not useful in associating extremely domain-specific concepts, but they made good suggestions when we tested general search concepts like "database management." The nature of the queries had great impact on the behavior and performance of the thesaurus activation module. In our testing, we simply assigned equal weights to the knowledge sources and the various link types. In reality, these weights will be determined by users and can thus provide better activation direction for our system. We plan to conduct a more complete evaluation of this module with both novices and subject experts.

WE ARE NOW MOVING OUR implementation from a 5-MIPS VAX/VMS 8600 development platform to a 25-MIPS DECstation 5000, and revising some search routines. After this conversion, we expect to achieve a better search response time.

We believe our study has generated insight for research and applications in heterogeneous knowledge-base integration and use. The blackboard architecture allows users to interface easily with knowledge bases and refine their queries incrementally. The spreading activation algorithm for the networks of weighted, labeled links suggests an exciting direction for creating active and "intelligent" document management systems. Findings from this study have significant impact on our recent work, which includes using automatic indexing and neural networks for concept classification of electronic meeting output;[16] developing an X-Windows interface for dynamic, Hopfield activation display; and designing an incremental, automatic thesaurus and a concept-based retrieval interface for a scientific database.

Acknowledgments

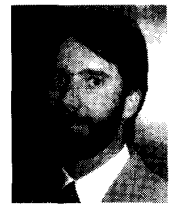
Hsinchun Chen's research was funded mainly by the National Science Foundation Research Initiation Award IRI-9211418 and the National Institutes of Health Biomedical Research Support Grant S07RR07002.

We obtained the computing-related terms and cross-references of the Library of Congress Subject Headings from the Online Computer Library Center in Dublin, Ohio.

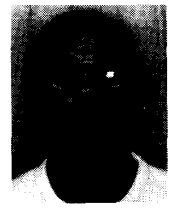
References

1. C.L. Foss, "Tools for Reading and Browsing Hypertext," *Information Processing and Management*, Vol. 25, No. 4, 1989, pp. 407-418.
2. D.L. Erman et al., "The Hearsay-II Speech Understanding System: Integrating Knowledge to Resolve Uncertainty," *ACM Computing Surveys*, Vol. 12, No. 2, 1980, pp. 213-253.
3. W.B. Croft and R.H. Thompson, "I^{3}R: A New Approach to the Design of Document Retrieval Systems," *J. Am. Soc. for Information Science*, Vol. 38, No. 6, 1987, pp. 389-404.
4. E.A. Fox, "Development of the Coder System: A Testbed for Artificial Intelligence Methods in Information Retrieval," *Information Processing and Management*, Vol. 23, No. 4, 1987, 341-366.
5. H. Chen and K.J. Lynch, "Automatic Contruction of Networks of Concepts Characterizing Document Databases," *IEEE Trans. Systems, Man and Cybernetics*, Vol. 22, No. 5, Sept./Oct. 1992, pp. 885-902.
6. M.J. Bates, "Subject Access in On-Line Catalogs: A Design Model," *J. Am. Soc. for Information Science*, Vol. 37, No. 6, Nov. 1986, pp. 357-376.
7. N.J. Belkin, R.N. Oddy, and H.M. Brooks, "Ask for Information Retrieval: Part I. Background and Theory," *J. of Documentation*, Vol. 38, No. 2, June 1982, pp. 61-71.
8. P.R. Cohen and R. Kjeldsen, "Information Retrieval by Constrained Spreading Activation in Semantic Networks," *Information Processing and Management*, Vol. 23, No. 4, 1987, pp. 255-268.
9. H. Chen and V. Dhar, "Cognitive Process as a Basis for Intelligent Retrieval Systems Design," *Information Processing and Management*, Vol. 27, No. 5, 1991, pp. 405-432.
10. R.K. Belew, "Adaptive Information Retrieval," *Proc. 12th Annual Int'l ACM/SIGIR Conf. on Research and Development in Information Retrieval*, ACM, New York, 1989, pp. 11-20.
11. D.E. Rose and R.K. Belew, "A Connectionist and Symbolic Hybrid for Improving Legal Search," *Int'l J. Man-Machine Studies*, Vol. 35, No. 1, 1991, pp. 1-33.
12. R.J. Brachman and D.L. McGuinness, "Knowledge Representation, Connectionism, and Conceptual Retrieval," *Proc. 11th Conf. Research and Development in Information Retrieval*, ACM, New York, 1988, pp. 161-174.
13. D.W. Tank and J.J. Hopfield, "Collective Computation in Neuronlike Circuits," *Scientific American*, Vol. 257, No. 6, Dec. 1987, pp. 104-114.
14. J.T. Schwartz, "Neural Networks and Artificial Intelligence," in *An Artificial Intelligence Debate: False Starts, Real Foundations*, S.R. Graubard, ed., MIT Press, Cambridge, Mass., 1989, pp. 85-121.
15. R.P. Lippmann, "An Introduction to Computing with Neural Networks," *IEEE ASSP Magazine*, Vol. 4, No. 2, Apr. 1987, pp. 4-22.
16. H. Chen et al., "Automatic Concept Classification of Electronic Meeting Output," to be published in *Comm. ACM*, 1993.

Hsinchun Chen is assistant professor of management information systems at the University of Arizona. His research interests include human-computer interactions, intelligent information retrieval, knowledge discovery in scientific databases, and neural-network computing. He received his PhD in information systems from New York University in 1989. He is a member of IEEE, ACM, AAAI, and The Institute of Management Sciences.



Kevin J. Lynch is the senior database systems engineer for Sherpa Corp. His research interests include distributed databases, collaborative research systems, and corporate intelligence systems. Lynch has consulted with and taught database applications development for several corporate clients. He holds a PhD and MS in management information systems and a BA in psychology from the University of Arizona in 1989. He is a member of ACM.



Koushik Basu is a doctoral student in management information systems at the University of Arizona. His research focuses on the applications of AI techniques in adaptive information retrieval for engineering and scientific applications. He received his BS in electronics and electrical communication engineering from the Indian Institute of Technology in 1987 and his MS in computer information systems from the University of Miami in 1990.

Tobun Dorbin Ng is a master's degree student in management information systems at the University of Arizona, where he received his BS in the same subject. His research interests include artificial intelligence, neural networks, and database systems.

The authors can be reached at the MIS Department, University of Arizona, Tucson, Arizona 85721; email, hchen@mis.arizona.edu