Knowledge-Based Document Retrieval: Framework and Design

Knowledge-based document retrieval: framework and design

Hsinchun Chen

University of Arizona, Tucson, AZ, USA

Received 24 August 1991
Revised 7 February 1992

Abstract. This article presents research on the design of knowledge-based document retrieval systems. We adopted a semantic network structure to represent subject knowledge and classification scheme knowledge and modeled experts' search strategies and user modeling capability as procedural knowledge. These functionalities were incorporated into a prototype knowledge-based retrieval system, Metacat. Our system, the design of which was based on the blackboard architecture, was able to create a user profile, identify task requirements, suggest heuristics-based search strategies, perform semantic-based search assistance, and assist online query refinement.

1. Introduction

Large electronic information storage and retrieval systems such as online catalogs, online bibliographic databases, customized electronic newspapers, legal and financial databases, and videotex are changing the way we gather, process, and retrieve information. These systems provide a wide variety of information and services, ranging from daily updates of foreign and national news, movie reviews, law cases, and financial data on companies to journal articles, books, trademarks, and statistics. However, gaining access to such information is often difficult. This is due in large part to the indeterminism involved in the process by which information is indexed and to the latitude searchers have in expressing a query. For inexperienced searchers, the problem of finding information relevant to their need can be difficult for three reasons:

(1) it can require a significant amount of knowledge of the subject area in which information is sought,
(2) it requires knowledge about the structure and functionality of the information storage and retrieval system, and
(3) it requires knowledge about the classification scheme or indexing language pertinent to the information storage and retrieval system.

Searchers may not know enough about the subject area for which answers are being sought, since the very purpose of the search is to acquire knowledge about the subject area. Searchers may have only a felt or conscious need, which requires to be formalized and refined [50,51]. A human information specialist such as a reference librarian often assumes an active role in helping searchers refine and articulate their queries.

"Intelligent retrieval systems" that aim to incorporate reference librarian-like capabilities into conventional document retrieval systems have attracted researchers from both information science and artificial intelligence. This article presents the implementation of such a system.

The article is organized as follows. In Section 2, we present a theoretical framework for understanding the information storage and retrieval process. Prior research and our own empirical studies are presented within this framework. In Section 3, we describe the structure and design of a prototype knowledge-based retrieval system (Metacat) that we developed. In Section 4, we outline a knowledge-based information retrieval process and we present an example for illustration. The implications of our research are discussed in the final section.

Correspondence to: H. Chen, MIS Department, College of Business and Public Administration, University of Arizona, Tucson, AZ 85721, USA.

Journal of Information Science 18 (1992) 293–314
Elsevier

2. A framework for document-based information retrieval

In this section, we present a theoretical framework for understanding the information storage

and retrieval process in terms of the "agents" involved in the indexing and searching processes. This framework, depicted in Fig. 1, shows the human agents involved in these processes, the types of knowledge these agents possess, and the observed characteristics of their indexing and searching behaviors. Prior research related to the components within this framework and findings from our own studies is presented in this section. Empirical findings regarding reference librarians' search strategies and cognitive process [11], searchers' misconceptions during information retrieval [13], and searchers' search strategies [14] have been reported in various publications. The focus of this article, however, is to present to readers a complete picture of a prototype knowledge-based system we developed, which was grounded on our empirical findings. We also discuss recent research that aimed at designing more "intelligent" and friendly document retrieval systems.

2.1. Knowledge components

Three types of knowledge are involved in online information retrieval. First, *classification scheme knowledge* that is used for indexing documents is also required to search for them. Secondly, *subject area knowledge* is required for expressing a query in appropriate terms. Lastly, *system knowledge* is necessary for operating effectively on specific online systems. These three knowledge components are presented towards the left of Fig. 1.

These knowledge elements have been reported in various information retrieval studies. Bates identified subject area knowledge and classification knowledge as factors that affect subject catalog search [4]. In a more detailed discussion, Belkin postulated that searchers exhibit an "anomalous state of knowledge" [7] regarding the subject area of their inquiries. The author of a document, on the other hand, has a much more

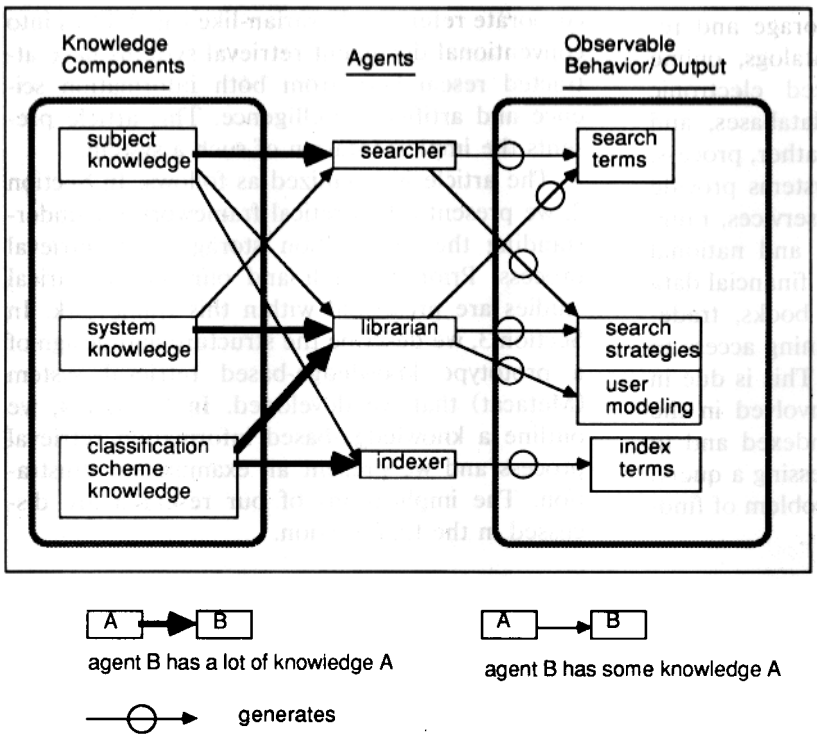

Fig. 1. A framework for information storage and retrieval.

complete state of knowledge concerning the subject area of the document. Chen and Dhar [13] identified the sources of problems that contributed to online document retrieval difficulties. They reported a taxonomy of subject area, classification scheme, and system errors that searchers made during online information retrieval. An online thesaurus was suggested to alleviate the subject area and classification scheme-related problems.

2.2. Agents

The three types of knowledge outlined above are typically distributed among three parties to whom we refer as "information agents." These agents include *indexers*, who classify the documents based on some pre-determined classification scheme; *searchers*, who express their queries using selected terms, and *reference librarians*, who serve as the intermediaries between searchers and retrieval systems.

Searchers generally do not have classification scheme knowledge. Their knowledge of the subject area and the system functionality varies widely. Indexers generally have a lot of classification scheme knowledge and some subject knowledge (they are, however, not involved in the retrieval process). Librarians must have all three kinds of knowledge, although they generally know more about the classification scheme and system than about various subjects. The relationships between the knowledge components and the agents are represented in Fig. 1 using links characterizing "strong" and "weak" knowledge.

2.3. Observable behaviour / output

Indexing uncertainty and search uncertainty are the primary sources of information retrieval problems. Indexing uncertainty arises because the different expert indexers can assign different index terms for a given document (see the box labelled "index terms" in Fig. 1). Search uncertainty arises because searchers have latitude in choosing terms to express a query (see the box labelled "search terms" in Fig. 1) and the search strategies they employ in acquiring information (see the box labelled "search strategy" in Fig. 1).

Because of the indeterminism involved in indexing and searching, an exact match between the searcher's terms and those of the indexer is unlikely. This is referred to as the *terms matching* problem. Secondly, the search terms used may not in fact represent what the searcher is really looking for (assuming for the moment that he knows what he wants). This is referred to as the problem of *query refinement*. In the remainder of this section, we discuss each problem in detail in the context of prior research.

2.3.1. Indexing uncertainty: index terms

The process of indexing is partly indeterminate. Evidence suggests that different indexers, well trained in an indexing scheme, might assign index terms for a given document differently. It has also been observed that an indexer might use different terms for the same document at different times [28,47].

Several approaches have been proposed by researchers seeking to improve the indexing of documents. These include: use of more sophisticated classification schemes such as the Dewey Decimal Classification [16], more extensive linkages between fields in different records that allow users to browse and navigate through a database [37], and the application of the "hypertext" concept to catalogs, that is, breaking the linearity of the traditional file structure and providing links in records in a variety of different directions [25].

In our research, we take indexing uncertainty as given. We focus only on improving the search process, assuming that some level of indexing uncertainty will always exist.

2.3.2. Search uncertainty: search terms, search strategies, and user modeling

Search uncertainty refers to the latitude searchers have in choosing search terms. There are many factors which may contribute to search uncertainty. We discuss them below.

(1) Search terms: A high degree of uncertainty with regard to search terms has been observed. Searchers tend to use different search terms for the same information. Studies have revealed that on average, the probability of any two people using the same term to describe an object is less than 20% [22,23,24]. This limits the success of various design methodologies for controlled vocabulary-driven interaction [22].

Bates [5] argues that for a successful match, the searcher must somehow generate as much "variety" (in the cybernetic sense, as defined by

Ashby [2]) in the search, as has been produced by the indexers in their indexing. The variety produced by an indexer can also be viewed as *redundancy* in the sense that it consists of partially overlapping classifications applied to a document. To increase the chances of a successful match, there should be a number of indexes for each document. This requires preserving the redundancy (generated by the indexer) associated with each document. In practice, however, catalog systems discourage redundancy for the following reasons [5,9]:

(a) *Whole document indexing*: The cataloger working according to the Library of Congress (i.e., Library of Congress Subject Headings, LCSH) or some other scheme is trained to index the whole document, not parts or concepts within it.

(b) *Uniform heading*: The principle of uniform heading holds that for any description there is to be one and only one heading which reflects that description.

(c) *Specific entry*: Each document is to be entered under a category (heading) which is specific to the content of the book, neither broader or narrower in scope than the scope of the book's contents.

(d) *Limited cross-reference structure*: cross references are frequently an afterthought to "augment" the basic catalog organization [3].

In summary, these tendencies to reduce redundant access points decrease the likelihood that a searcher will generate the right term for retrieval. Recognizing this problem, reference librarians often rely on extensive thesaurus consultation and searcher query refinement in an attempt to generate the "variety" associated with the search terms and to increase their chance of matching index terms.

While indexers use the rule of specificity for indexing, searchers tend to approach a search by specifying broader terms first. There might be several reasons for this. One hypothesis is that searchers often do not have "queries", but what Belkin calls an "anomalous state of knowledge" [7]. Searchers often expect to refine this anomalous state into a query *through* an interactive process. The organization of a catalog or a system, however, does not always facilitate this type of query refinement. In contrast, reference librarians appear to be particularly adept at this function. Taylor suggests that a searcher's queries start from an actual but unexpressed need (visceral need). The visceral need is refined to a conscious description of the need (conscious need). This need is finally formalized as a statement (formalized need). The actual query presented to the information system, however, may be compromised due to the searcher's expectation of the system (compromised need) [51]. Based on Taylor's model, a similar model for describing query refinement during the pre-search interview between the reference librarians and the online searchers was developed by Markey [30]. Chen identified a similar process of query refinement in an online retrieval setting in which searchers formalized and/or compromised their information needs during online search [12]. The importance of query refinement during the information retrieval process is well documented in the prior research.

(2) Search strategies: Despite the fact that considerable latitude is involved in selecting the term a searcher may employ to describe a subject area, the approach adopted by the searchers for performing a search varies. Search strategy usually refers to a plan or approach for the whole search, whereas search tactic refers to a move or maneuver made to further a search [4]. Bates has described 29 tactics that are used in information searching. These tactics are grouped into four categories: (1) monitoring tactics are actions to keep the search on track, (2) file structure tactics are techniques for traversing the information within the information system, (3) search formulation tactics are tactics to aid in the process of designing or redesigning the search formulation, and (4) terms tactics are moves to select and revise search terms. These tactics are applicable in both manual and online systems. In a card catalog study, two strategies for searching have been identified: a "self-reliant" style where searchers generate their own search terms and a "catalog-oriented" style where searchers use the terms found in the card catalog [49]. Bourne identified two search strategies. In the "building-block" strategy, the user enters various terms as separate search statements. After the search results are derived, he or she combines all search statements into a single final statement using the Boolean operator, AND. This strategy

contrasts with the "pearl-growing" strategy, in which the user initially searches a few specific terms to retrieve some citations. These citations are then examined for new candidate search terms to be added to subsequent searches [31]. These two strategies were also verified in Palmer's study [38]. Chen and Dhar [14] reported observing five online search strategies adopted by searchers. Two strategies, which are based on trial-and-error and forward–backward screen browsing, respectively, were adopted by inexperienced searchers and were generally ineffective. Three strategies which were based on extensive thesaurus consultation, utilization of efficient online options, and exploration of known citations, were adopted by various experienced searchers and reference librarians. Thesaurus consultation and known citations exploration can be considered as variants of the building-block and pearl-growing strategies, respectively.

There have been many attempts to capture information specialists' domain knowledge, search strategies, and query refinement heuristics in document retrieval systems design. We summarize some important system development works here.

Coalsort [34], a knowledge-based system, facilitates the use of bibliographic databases in coal technology. A semantic network, representing an expert's domain knowledge, embodies the system's intelligence. GRANT [17], developed by Cohen and Kjeldsen, is an expert system for finding sources of funding for given research proposals. Its search method—constrained spreading activation in a semantic network—makes inferences about the goals of the user and thus finds information that the user did not explicitly request but that is likely to be useful. Shoval [45] developed an expert system for suggesting search terms. This system is composed of two components: the knowledge base, represented as a semantic network in which the nodes are words, concepts, or phrases. Links express the semantic relationships between the nodes. The second component is made up of rules, or procedures, which operate upon the knowledge base and are analogous to the decision rules of the information specialist. Fox's CODER system [21] consists of a thesaurus that was generated from the *Handbook of Artificial Intelligence* and *Collin's Dictionary*. In CANSEARCH [39], a thesaurus is presented as a menu. Users browse and select terms from the menu for their queries. The "Intelligent Intermediary for Information Retrieval" (I^3R), developed by Croft [18], consists of a group of "experts" that communicate via a common data structure, called the blackboard. The system consists of 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. This blackboard architecture, which is derived from the HEARSAY-II system [20] has also been used by Belkin [8,6] and Fox [21] for designing retrieval systems. The IOTA system, developed by Chiaramella and Defude, includes natural language processing of queries, deductive capabilities (related to user modeling, search strategies definition, use of expert and domain knowledge), management of full-text documents, and relevance evaluation of answers [15].

The National Library of Medicine's thesaurus projects are probably the largest-scale effort that uses the knowledge in existing thesauri. In one of the projects, Rada and Martin [40,32] conducted experiments for the automatic addition of concepts to MESH (Medical Subject Headings) by including the CMIT (Current Medical Information and Terminology) and SNOMED (Systematized Nomenclature of Medicine) thesauri. Access to various sets of documents can be facilitated by using thesauri and the connections that are made among thesauri. The Unified Medical Language System (UMLS) 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 [27,33,29]. The UMLS 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.

(3) User modeling: Another important component during information retrieval is the user modeling (or *patron* modeling) capability, which is unique to reference librarians. During the user–librarian consultation process, the librarian develops an understanding of the type of user being dealt with on the basis of verbal and non-verbal clues. Usually, the educational level of the user, the type of question, the way the question is phrased, the purpose of the search, and the expected search results all play a major role in helping the librarian determine the needs of the user. The librarian, in essence, creates models of the user profile and the task requirement during the consultation process.

User modeling has played a crucial role in applications such as question-answering systems, intelligent tutoring systems, and consultation systems [1,48,46,53,11]. An intelligent interface for document retrieval systems must also exhibit the user modeling capability of experienced human intermediaries. Daniels proposed a frame-based representation for a user model and rules for interacting with the users. She has shown that user modeling is a necessary function in the presearch information interaction [19]. Rich's Grundy system builds models of its users, with the aid of stereotypes, and then uses those models to guide it in its task, suggesting novels that people may find interesting [41,42,43].

In the next section, we present an "intelligent" document retrieval system we developed, based on our empirical studies and prior research. Our design was aimed at addressing the problems and issues discussed in our information retrieval framework.

3. A knowledge-based design for document retrieval systems

Our system resembles I^3R and IOTA in its attempts to assist users during all stages of information retrieval using various system-supported procedures and aids. Specifically, our system consists of a user and task modeling module, a heuristics-based search strategist, an automatic browser for the knowledge base (which is constructed as a semantic network), term and document ranking mechanisms, and an interface soliciting the user's relevance feedback. The system design is mainly based on the *blackboard* architecture. Our system differs from other knowledge-based retrieval systems in its extensive use of the users' and the information specialists' knowledge and heuristics, its attempts to alleviate the user's misconceptions during the search, its adoption of the strengths of the existing inverted index-based systems, and the utilization of a uniform and coherent blackboard architecture. We will discuss the unique features of our system after a thorough discussion of it.

The first blackboard system was the HEARSAY-II speech understanding system [20] that evolved between 1971 and 1976. Subsequently, many complex knowledge-based systems that adopt similar architecture have been built. (Readers are referred to [36] and [35] for a good overview of the blackboard architecture and blackboard-based systems.) Typically, a blackboard structure consists of three components:

(1) *The knowledge source*: The knowledge needed to assist the system users is partitioned into different knowledge sources, which are kept separate and independent. Each knowledge source acts as a small expert.
(2) *The blackboard data structure*: The data involved in the problem solving process are kept in a global database, the blackboard. Knowledge sources produce changes to the blackboard that lead incrementally to a solution to the problem. Communication and interaction between the knowledge sources take place solely through the blackboard.
(3) *The control module*: This module controls the sequence of operations of the system. It monitors changes on the blackboard and selects the appropriate knowledge sources.

Although our system is similar to blackboard systems in its architecture and components, it does not emulate the *opportunistic problem solving* in HEARSAY-II nor its extensive use of explicit knowledge sources (e.g., word spotting, phrase-island generation, phrase extending, etc.).

Our system also incorporates three blackboard system components. First, five types of data are included in the blackboard. They are: user model, task model, query model, index terms that correspond to the searcher's query, and citations that are derived from the search. While the first three types of data reveal the searcher's information need, the last two represent search results. Sec-

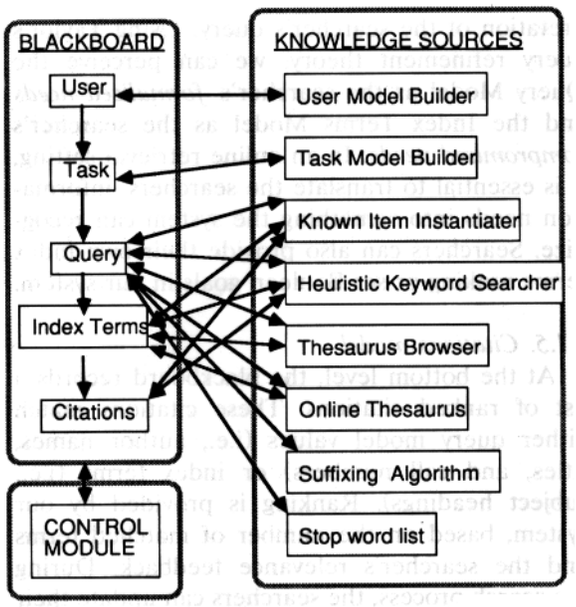

Fig. 2. The blackboard architecture of Metacat.

ondly, we have incorporated eight "knowledge sources" (individual system modules) into our system. They are: the user model builder, the task model builder, the suffixing algorithm, the stop word list, the online thesaurus, the known item instantiater, the heuristic keyword searcher, and the thesaurus browser. Lastly, we have also included several data-driven inferencing rules in the control module. These components are shown in Fig. 2 and are presented in chronological order.

3.1. The blackboard

The five data types posted on the blackboard are represented as *frames*, each consisting of a few attributes (slots). The five data types in the blackboard are organized as a hierarchy to depict their specificities to the searcher's information need. At the highest level, a user model is built, which captures the long-term characteristics of the searcher. At the second level, a task model is used to represent the searcher's task-related information, such as the type of material, the recency of publication, and the number of relevant citations the searcher desires. At the third level, the system creates a model of the actual query, which includes all the query-related inputs supplied by the searcher such as search terms, author names, titles, etc. At the fourth level, index terms are generated by the system to represent the searcher's query. At the fifth and last level, citations retrieved during the search process are posted on the blackboard. These actual citations represent possible answers to the searcher's query. The goal of the system is to generate a reasonable set of relevant citations for the searcher. The status of the data posted on the blackboard will affect the selection of the various knowledge sources.

3.1.1. User model

The user model captures the long-term characteristics of the searcher. The attributes of this frame-based model include: the searcher's identification (a unique identifier, e.g., Social Security number or student id.), the searcher's educational field, the searcher's familiarity with the LCSH indexing scheme, the searcher's familiarity level with the subject area, and the data the model is created. We assigned default values to the three familiarity level attributes: searchers in general have low system familiarity, low LCSH familiarity, and medium subject area familiarity. These three attributes serve to represent a searcher's general level of understanding in the three knowledge areas discussed earlier in Section 2. However, the searchers' educational field and educational level may change these default values. Searchers' long-term characteristics may also affect their task-related requirements, such as: the recency of the information that is deemed appropriate, the precision and recall level that the searchers may want, etc. We will discuss this in the user model builder subsection. Once this model has been created (after the user has used the system once), it can be stored in the system. The system retrieves a searcher's model when the searcher uses the system again.

3.1.2. Task model

The task model captures the searcher's information need. The attributes of this model include: the purpose of the search (e.g., class reading, dissertation, etc.), the type of material the searcher wants (e.g., book, journal article, etc.), the number of citations the searcher wishes to get, and the recency of the publications that the searcher requests.

The task model reveals the searcher's expected search outputs. It is created early in the search process—right after soliciting the user-related information. During the search, however, searchers can change their information requirements, e.g., change the number of expected documents, recency of publications, etc. Our system will then use the new task model information to guide the search process.

3.1.3. Query model

In the third level, the search inputs provided by the searcher are recorded in a query model. The attributes in this model include; author names, book titles, call numbers, and search terms,—all supplied by the searcher. This information, which reveals the searcher's query, is used by our system to generate corresponding index terms and citations.

3.1.4. Index terms model

The terms in the query model are used to generate index terms that the system can recognize. While the query model reveals the searcher's query, index terms represent the system's interpretation of the searcher's query. Using Taylor's query refinement theory, we can perceive the Query Model as the searcher's *formalized needs* and the Index Terms Model as the searcher's *compromised needs*. In an online retrieval setting, it is essential to translate the searchers' information needs into something the system can recognize. Searchers can also provide their own index terms ranking or set Boolean goals in our system.

3.1.5. Citations model

At the bottom level, the blackboard records a list of ranked citations. These citations match either query model values (i.e., author names, titles, and call numbers) or index terms (i.e., subject headings). Ranking is provided by our system, based on the number of matched terms and the searcher's relevance feedback. During the search process, the searchers can update their selected citations. It is likely that as the searchers' queries evolve and their search results accumulate, they may decide to delete some old selections or add new ones.

In Fig. 3, we list the five blackboard data types and their associated attributes. The domains and

```
(USER:
      user_id: (number)
      edu_level: (literal)
      edu_field: (literal)
      subject_familiarity: (low, medium, high: MEDIUM)
      indexing_familiarity: (low, medium, high: LOW)
      system_familiarity: (low, medium, high: LOW)
      date_created: (literal)  )
(TASK:
      user_id: (number)
      type_of_material: (literal)
      purpose_of_search: (literal)
      #_of_hits_expected: (number)
      recency_of_publication: (number)
      expected_recall_level: (low, medium, high: MEDIUM)
      expected_precision_level: (low, medium, high: MEDIUM))
(QUERY
      authors: (list of literals)
      titles: (list of literals)
      call_numbers: (list of literals)
      search_terms: (list of literals)  )
(INDEX_TERMS:
      selected_index_terms: (list of literals)
(CITATIONS:
      selected_citations: (list of literals)
```

Fig. 3. Representation of the blackboard objects.

default values (indicated by the upper case terms in the attributes) of these five data types are also shown in this figure.

3.2. The knowledge sources

Our system consists of eight knowledge sources. Each knowledge source is capable of performing specific functions. We examine each knowledge source in order.

3.2.1. User model builder

The user model builder assumes two functions in the search process. First it creates a long-term model of the user by soliciting the user's id. number, educational level, and educational field. Secondly, based on this information, it infers the searcher's familiarity levels in the subject area, the classification scheme, and the system. These inferencing heuristics are represented as production rules (IF ... THEN ... rules).

The default value for the subject familiarity level is medium. For searchers who are freshmen or sophomores, the subject familiarity level is changed to low. Searchers at the graduate level will be assigned a high subject area familiarity level. Only searchers who are Information Sciences majors will be assigned a high indexing familiarity level. The default value is low. The default system familiarity level is set as low initially. After a searcher uses the system a few times, this value will be increased to medium or high based on the number of times and the frequency with which the searcher uses Metacat. The information in the user model and the task model will affect the number of citations and the recency of publications generated by the system.

3.2.2. Task model builder

The task model builder also performs slot-filling and heuristics instantiation. In the slot-filling process, it solicits the user's information concerning the type of material the searcher wants, the purpose of the search, the number of citations expected, and the recency of the publications desired by the searcher. In the heuristics instantiation process, the system applies its task-related heuristics. We identified three types of heuristics from our empirical study of university library searchers [14]. We refer to them as *consistency checking*, *LCSH instantiation*, and *recall / precision instantiation* heuristics.

(1) *Consistency checking heuristics*: These heuristics ensure that the type of material the user wants and the purpose of the search are consistent with the user's background. Queries that are questionable may involve freshmen or sophomores looking for citations for a dissertation or a research article (purpose of search) or proceedings articles (type of material). In our empirical study, reference librarians frequently stressed the importance of an early consistency check before starting an actual search. As reported in [11], reference librarians often perform *stereotypical user modeling* during consultation. They believe users with a higher level of subject familiarity (e.g., PhDs) are likely to require more academically oriented in-depth information. In contrast, users with a lower level of subject familiarity are likely to require less scholarly, more general material.

(2) *LCSH instantiation heuristics*: The type of material the user desires may affect the selection of the subject headings (LCSH terms). For example, if the type of material the user wants is an edited book or proceedings, then the librarian might include "Congresses" as a subdivision of an index term, e.g., "Psychology-Congresses." If the type of material requested is a journal, then the subdivision would be "Journals," e.g., "Artificial Intelligence—Journals." These heuristics represent the system's LCSH knowledge.

(3) *Recall / precision instantiation heuristics*: These heuristics help determine the number of citations to obtain for the user. We have developed a few rules to determine the search's expected *recall* and *precision* levels. *Recall* is defined as the portion of relevant citations retrieved by a searcher from the database. It indicates the *completeness* of a search. *Precision* is defined as the portion of retrieved citations that are found to be relevant to the user's information need. It indicates the *preciseness* of a search. Expected recall and precision levels of a search can be determined by the searcher's educational level and the purpose of search. For sophisticated users (e.g., faculty members or PhD students) working on more academically oriented tasks (e.g., a dissertation or research paper), the expected recall level for the search should be high (because of the need for complete references). In contrast,

for freshmen or sophomores working on class papers or reading, the expected recall should be low (since they often do not require too much information), but the precision should be high (because they are less able to judge the relevance of information). The system-generated expected recall and precision levels in turn determine the "appropriate" number of citations for the searcher.

We have developed the following *ad hoc* rules to compute the number of system-generated citations. If the number of citations the user wants is x, and the expected recall level for the search is low, medium or high, respectively (derived from the recall/precision instantiation heuristics), then the number of citations the system should come up with is $0.5x$, x, or $2x$, respectively. If the number of relevant citations the user wants is x, and the expected precision level for the search is low, medium or high, respectively, then the number of citations the system should generate is $2x$, $1.5x$, or x, respectively. (The choice of these numbers is quite arbitrary. They can be changed to reflect the condition of a retrieval environment.) For example, for a search that requires high recall and low precision, possibly a faculty member looking for citations for a research paper, our system will try to find four times the number of citations requested (i.e., 2 * 2). Because our system identifies only potentially relevant citations, noise often exists. Searcher relevance feedback is then needed to identify the relevant citations.

These heuristics are important in restricting our system's search efforts. Our system terminates its search when it finds the desired number of citations. Currently, our system contains about 20 rules in its user model builder and task model builder.

3.2.3. *Suffixing algorithm*

While the user model builder and the task model builder represent the searcher's profile and request, the suffixing algorithm and the stop word list support the system's search functions. The suffixing algorithm first identifies the root form (the stem) of the input word and then generates a list of legal suffixed words from the root word. The suffixing algorithm increases the chance of matching search terms with index terms. It helps resolve the *terms matching* problem discussed earlier.

The suffixing component consists of two parts. First, it includes a 28,000 word (root words) dictionary with flags indicating legal suffixed forms. The total number of words our system can recognize is about 80,000. Secondly, there are about 30 rules to interpret the flags for suffixes. The suffixes created by this algorithm include: ive, ion, tion, en, ions, ications, ens, th, ieth, ly, ing, ings,

```
* and @ -- ''variables'' that can stand for any letter.
Upper case letter -- constants.
''...'' -- any string of zero or more letters.
.eq. -- equal.
.ne. -- not equal.
---------------

''V'' flag:
      ...E --> ...IVE as in CREATE --> CREATIVE
      if * .ne. E, ...* --> ...*IVE as in PREVENT --> PREVENTIVE
''N'' flag:
      ...E --> ...ION as in CREATE --> CREATION
      ...Y --> ...ICATION as in MULTIPLY --> MULTIPLICATION
      if * .ne. E or Y, ... -> FALLEN
''X'' flag:
      ...E --> ...IONS as in CREATE --> CREATIONS
      ...Y --> ...ICATIONS as in MULTIPLY --> MULTIPLICATIONS
      if * .ne. E or Y, ...* --> ...*ENS as in WEAK --> WEAKENS
```

Fig. 4. Examples of suffixing rules.

ed, est, er, ers, s, es, ies, ness, iness, and 's. A few sample rules are shown in Fig. 4. For example, the dictionary entry "CREATE" has three flags "V," "N," and "X" to indicate its legal suffixed forms "CREATIVE," "CREATION," and "CREATIONS."

3.2.4. Stop word list

Words in a query which do not bear semantic content are included in a stop word list in our system. One of the search engines, the heuristic keyword searcher, uses this stop word list to filter the searcher's input. It removes stop words from searchers' queries when performing single-word search. Words in the stop word list include: prepositions (e.g., on, in, at, etc.), pronouns (e.g., she, he, I, etc.), auxiliary verbs (e.g., can, would, could, will, etc.), etc. We have about 160 words in our stop word list.

3.2.5. Online thesaurus

The next knowledge source is the *online thesaurus*, which represents the subject area and classification scheme knowledge. This knowledge source is represented in the form of a semantic network-based online thesaurus. We constructed the semantic network by extracting a portion of the computer readable form of the LCSH Handbook (with the assistance of OCLC). Our online thesaurus consists of nearly 3,500 terms (both official terms and unofficial terms) in the areas of mathematics and computer science, which were the areas we used for evaluating the system. Each term has between a couple of, and a few hundred, relevant terms associated via the cross referencing structure of the thesaurus. The thesaurus was implemented in FLAVORS (an object-oriented language). Since the complete LCSH is in an online format, the process of generating a semantic network representation requires no manual effort.

Each semantic network node is represented as a FLAVORS frame with the following attributes: (An example is shown in Fig. 5.)

(1) Term: Specify the name of the term.
(2) Type of term: Specify whether the term is an official subject heading or an unofficial term.
(3) NT: Specify all narrower official terms.
(4) BT: Specify all broader official terms.
(5) RT: Specify other related official terms.
(6) USE: Specify the synonymous official terms. This link relates unofficial terms and official terms.

```
Term Object Frame:
     (Term: (name of the term)
          Type of term: (* for unofficial term; nil for official term)
          NT: (list of narrower terms)
          BT: (list of broader terms)
          RT: (list of related terms)
          USE: (list of synonymous official terms)
          UF: (list of synonymous unofficial terms)
          Number of citations (if-needed): (integer)
          Matched citations (if-needed): (list of matched citations)


  Example of Term Instance:
     (Term: computers
          Type of term: nil
          NT: (electronic-data-processing fifth-generation-computers)
          BT: (machine-theory)
          RT: (calculators)
          USE: nil
          UF: (electronic-computer electronic-brain)
          Number of citations (if-needed): 56
          Matched citations (if-needed):
              (''Introduction to Computers'' ''Computers''..)
```

Fig. 5. Frame-based representation for the knowledge elements.

(7) UF: Specify the synonymous unofficial terms.
(8) Number of matched citations: Specify the number of citations indexed under the official term. It is computed when the user so requests (i.e., an *if-needed facet* in the frame representation [52]).
(9) Matched citations: Specify the matched citations. Users can examine this list of citations to check the relevance of the citations to their queries. This, again, can be computed when the user so requests (another *if-needed facet*).

The online thesaurus is a *passive* knowledge source that needs to be activated by other procedural knowledge sources such as the *thesaurus browser*.

3.2.6. Known item instantiater

Our system consists of three search engines that simulate human search strategies. The first search engine executes the *known item instantiation* strategy. Searchers first identify some relevant citations through the use of the conventional known item search options (e.g., author, title, and call number search). The system then uses the subject headings assigned to these retrieved citations to perform a subject search. This in turn generates more citations that can be identified by searchers. This strategy can be applied repeatedly. When a new citation is derived, the system can obtain a few new subject headings. These new subject headings may lead to other new citations, which in turn may suggest other new subject headings. By following these links, we can identify a set of relevant subject headings and citations. The usefulness of this *relevance feedback* search method has been well documented in prior studies [44].

This search engine initially relies strongly on the existing known item search options. It then follows its subject heading-citation activation continuously. The computation involved in this process is relatively "inexpensive," compared with the next two search engines. By using the existing search options wisely, an online search process can become more productive and efficient.

3.2.7 Heuristic keyword searcher

The *heuristic keyword searcher* is another search engine which utilizes the existing search options in a new way. While the known item instantiater exploits the known item search options, the heuristic keyword searcher utilizes the non-known item search options, in particular, the (complete) subject search (SUB, which performs exact matches), the keyword subject search (SUBK, which performs partial matches), the (complete) title search (TIL, which performs exact matches) and the keyword title search (TILK, which performs partial matches). SUB, SUBK, TIL, and TILK are options available in most online catalogs.

This heuristic keyword searcher consists of ten different search functions. Each search function exhibits a different level of *credibility* in generating index terms that are semantically close to the search terms. When the system invokes this search engine, it first tries the most credible search function. If it fails, then the system proceeds to the next most credible search function. This process continues until the system either finds some index terms by using a certain search function or has exhausted all ten search functions. We discuss each of them here.

(1) SUB with the original search term: The system uses the original search term in subject search (SUB) to match some existing index terms. This is the most credible option because if there is a match, the index term would completely represent the search term.
(2) SUB with the suffixed search term: If there is no match from the first function, this function applies the suffixing algorithm to add suffixes to each word in the search term. For example, if there is no match using "online computer," then the system will try, "online computers," "online computing," etc.
(3) Thesaurus lookup with the original search term: If the search term does not match with any index terms in the thesaurus using the two functions described above, the system expands its search space by searching the unofficial terms the system recognizes in the online thesaurus. If the system matches an unofficial term in the thesaurus, it can obtain the synonymous official term by following the USE link in the online thesaurus. This function is about as reliable as the previous two functions because of the characteristics of the synonymous cross reference.
(4) Thesaurus lookup with the suffixed search term: Similar to the second function, if there is no match of the original term, our system

uses the search term's suffixed forms in thesaurus lookup.

(5) SUBK with the original search term: If our system finds no match by using subject search and thesaurus lookup (the previous four functions), it proceeds to perform a keyword subject search (SUBK) with the original search term. Unlike SUB which requires an exact match between the search term and the index term, SUBK generates all partially matched index terms for a search term (for example, a search term like "analytical" will generate index terms like "analytical hierarchy method," "analytical analysis," etc.). Because the keyword search options can find partial matches for the search term, the system's search space is greatly expanded. However, the matched terms derived would be less reliable than those generated from the previous functions.

(6) SUBK with the suffixed search term: As in performing the second function, our system expands its search space by using the search term's suffixed forms in a keyword subject search when there is no match from the previous functions.

(7) SUBK with the suffixed words derived from the search term: When the search term (which often has multiple words) generates no match, the system uses the suffixed *words* that are derived from the search term to perform SUBK search. For example, our system would first generate the suffixed words "online", "computer", "computers", and "computing" from the search term, "online computer". These words will then be used in SUBK. This process of decomposing the search term into suffixed search words can expand the system's search space significantly.

(8) TILK with the original search term: As was observed in our empirical study, title keyword search can assist searchers in performing subject-based search. After obtaining matched citations by using title keyword search, our system can then elicit subject headings from these matched citations. For online catalog records, which typically have little content information, title is a good source from which to glean the contents of the book. It is, however, less credible than searches using the subject heading directly (i.e., SUB, thesaurus lookup, and SUBK).

(9) TILK with the suffixed search term: Again, we can expand the search space by using the suffixed search term.

(10) TILK with the suffixed words derived from the search term: An even broader search space can be obtained by using the individual suffixed words derived from the search term.

The rationale behind this search engine is that the system expands its search space only when it is necessary. As stated in the design principle of online catalogs proposed by Bates [5], it is crucial for the searcher to get into the system in the first place—what she described as the (hit the) "side-of-the-barn" principle. This principle suggests a retrieval system that creates a "big target" for the searcher to hit. Our heuristic keyword searcher is grounded on this principle. Moreover, instead of merely presenting a big target to the searcher, our system suggests an incrementally expanded target. That is, the heuristic keyword searcher expands its search space incrementally. This "search expansion" process not only helps identify good index terms but is also computationally efficient.

3.2.8. Thesaurus browser

During a search, the system first invokes the known item instantiater and the heuristic keyword searcher. If the user is not satisfied with the results derived from these two search engines, the system automatically activates the thesaurus browser. The *thesaurus browser* uses our semantic network-based knowledge base (online thesaurus).

The index terms generated from these two search strategies can match some of the nodes (terms) in our online thesaurus. Matching nodes are taken as *source nodes* by our thesaurus browser. Our system applies a *heuristic spreading activation* process on the semantic network to generate relevant terms. This process is similar to the spreading activation process implemented in Grant [17], but differs from it in relying on a branch-and-bound algorithm to choose activation paths. Our system activates new nodes (terms) by following the links leading to/from the source nodes and applies the following heuristics to guide the activation process.

(1) *The Specific Terms First Heuristic*: Based on the analysis of the LCSH structure, we observed that nodes (terms) which have fewer neighbors in the semantic network are generally more specific (in content) than nodes (terms) which have more neighbors. Since users have a tendency to state their information needs more broadly than they should (as described earlier), our system applies a heuristic which visits nodes having fewer neighbors (the more specific terms) before it visits nodes with more neighbors.

(2) *The Specific Links First Heuristic*: Links associated with official terms are of three types: NT, RT, and BT. The system adopts a heuristic for expanding the links in the order of: NT, RT, and BT. That is, it traverses the NT links before it traverses the RT links, and before it traverses the BT links. This heuristic, which suggests search on specific links will lead to the activation of more specific terms.

(3) *The Shorter Distance First Heuristic*: This heuristic is related to the distance between an activated node and the source nodes. During the activation process, the system will expand nodes which are closer to the source nodes (shorter distance) earlier than those nodes which are further away from the source nodes (longer distance). The rationale is that, terms which are more remote (from the source nodes) are less relevant to the source nodes than terms which are closer to the source nodes. Therefore, they should be expanded only after the more relevant terms (closer nodes) have been expanded.

(4) *The Two Level Expansion Heuristic*: As described earlier, the number of links between two nodes (the distance between them) in a semantic network determines the semantic proximity of the nodes. In order to find only terms which are closely relevant to the source terms, the system expands each source node by only two levels. That is, we activate only nodes that are two links away from the source nodes. Because each node in the network may have between a few dozen and a few hundred links, even two-level expansion may require a lot of computation. This two-level expansion heuristic was derived from the users' query refinement pattern—they rarely expand the specificity level of their search terms by more than two levels, and an analysis of the cross referencing structure of LCSH. This heuristic helps ensure that our system finds terms that are semantically close to the source nodes (terms).

These four heuristics, which consider the specificity of the nodes, the specificity of the links, the distance between nodes, and the expansion level, are used to direct the system's *spreading activation* effort. We perceive spreading activation as a *search* problem in which the goal is to find the shortest path to each of the source nodes, with the total distance (a *relevance distance*) computed, based on the nodes visited, the types of links traversed, and the number of links in the paths. We developed a formula for the *relevance distance* calculation, which assign weights and costs to links and nodes, respectively. (Algorithmic details are presented in [10].) The *relevance distance* computation is sketched below:

```
(relevance distance accumulated so far) *
(relative weight on the next link) *
(number of neighbors of the next node)
```

We developed a branch-and-bound algorithm for spreading activation that is similar to the branch-and-bound search described in [52]. The initial *relevance distance* assigned to each source node is equal to its number of neighboring nodes. The system also assigns relative weights to the three types of links—the NT link has a lower weight than the RT link, which in turn has a lower weight than the BT link. Starting from some source nodes, our system will visit those neighboring nodes which are connected by the NT link and which have fewer neighbors sooner before visiting neighboring nodes which are connected by BT links and which have many other neighbors. Every partially-explored path has a *relevance distance* assigned to it and all paths are organized in a queue in sorted order with the shortest partial path listed at the front of the queue. The shortest path is then expanded and the queue is re-sorted based on the *relevance distance* associated with the new paths.

This branch-and-bound algorithm helps our system efficiently find other relevant and often more specific terms in the online thesaurus, a process akin to a reference librarian's thesaurus consultation. The branch-and-bound process terminates when all source nodes have been con-

nected or when it has completely activated all nodes that are two links away from the source nodes (the *two level expansion heuristic*).

After the activation process, our system "chunks" the expanded paths and the associated terms into different *concept groups*. The expanded paths that are linked together (i.e., *connected components* in Data Structures and Algorithms [26]) are considered to be in the same *concept group*. The nodes (terms) on the paths within the same concept group may address a similar underlying concept. This concept grouping function was observed as being sometimes adopted by reference librarians to identify the different sets of topics that users tried to address during a consultation session. New terms (terms which are different from the source terms) found in each concept group become good candidate terms for the user's queries. Our system ranks the concept groups in order based on the number of source terms involved in each concept group and suggests them to the user. User relevance feedback then follows.

In a test case for a query with three starting terms, this browsing process explored 345 paths in about 15 seconds and suggested 10 relaxed index terms. For queries with less than 10 starting terms, the response time for this thesaurus-browsing process was usually less than 30 seconds. The four search heuristics we incorporated are crucial to the system's search performance. An earlier search algorithm which did not include the heuristics-based branch-and-bound algorithm performed very poorly. For the same test case, it took about 25 minutes and it generated too many irrelevant index terms (a few hundred).

3.3. The control module

A control module determines the conditions for activating the knowledge sources, especially the three search engines—other knowledge sources are called upon by the control module to support the three search engines. The general principle behind the control module is to try simpler (less computationally expensive) search strategies first before attempting more search-intensive methods. The known item instantiation strategy utilizes information associated with some known documents. Its computation is relatively simple. The heuristic keyword search performs various keyword search operations, suffixing, and thesaurus lookup, which require more computation than known item instantiation. The heuristic keyword search, however, is less computationally intensive than the thesaurus browsing process, which involves activating numerous terms and links in the online thesaurus.

Table 1
The control module

	Next strategy to use	
		No doc. found.
thesaurus browser		
heuristic keyword		
known item inst		

Based on the search results obtained from the search strategy just used, our scheduler decides which strategy to use next. For example, if some new relevant documents are found by using any of the three search strategies (see Columns 1 and 2 in Table 1), our system goes back to the known item instantiation strategy immediately in an attempt to utilize the information embedded in the new documents. Also, during initial interaction with our system, our system prompts searchers to perform the known item instantiation search first. That is, our system asks searchers to come up with some known documents via the author, title, and call number search first before using other search approaches.

As indicated in Table 1, if no document can be identified by applying the known item instantiater, the least complex search engine, our system automatically activates a more complex and powerful search strategy, the heuristic keyword search. Similarly, if no document can be obtained by using the heuristic keyword searcher, our system will activate the thesaurus browsing module. Finally, if the thesaurus browser cannot help identify any relevant document, our system performs the heuristic keyword search repeatedly on the terms derived from the thesaurus. This process continues until either some relevant documents are identified (which will move the control module back to the regular mode), or until searchers decide that nothing relevant can be found and abandon their search.

4. A knowledge-based information retrieval process

The previous section attempted to explain the overall architecture of the system and the functionality of each system component. In this section we present an overview of the information retrieval process using a prototype system we developed. At the end of this section we also present an example for illustration. The flow of information processing adopted in our system is shown in Fig. 6.

The eight knowledge sources we discussed can be divided into three categories. The first category consists of the user model builder and the task model builder (indicated by the box at the top of Fig. 6). They assume the function of constructing a model of the searcher's profile and task requirements. This process is performed at the beginning of a search session. During the search, searchers can modify this model to reflect their changes of task requirements (see the arrow from Relevance Judgement to Task Model Builder in Fig. 6), which then will be considered by the system's search engines.

The second category contains the knowledge sources that play a supporting role for the search engines. These knowledge sources include: the stop word list, the suffixing algorithm, and the online thesaurus (shown in the L-shaped area at the lower left hand corner of Fig. 6). These knowledge sources are used to enhance the search capability of the heuristic keyword searcher, and the thesaurus browser.

The last category, shown at the center of Fig. 6, includes three search engines (the known item instantiater, the heuristic keyword searcher, and the thesaurus browser) and a relevance judge-

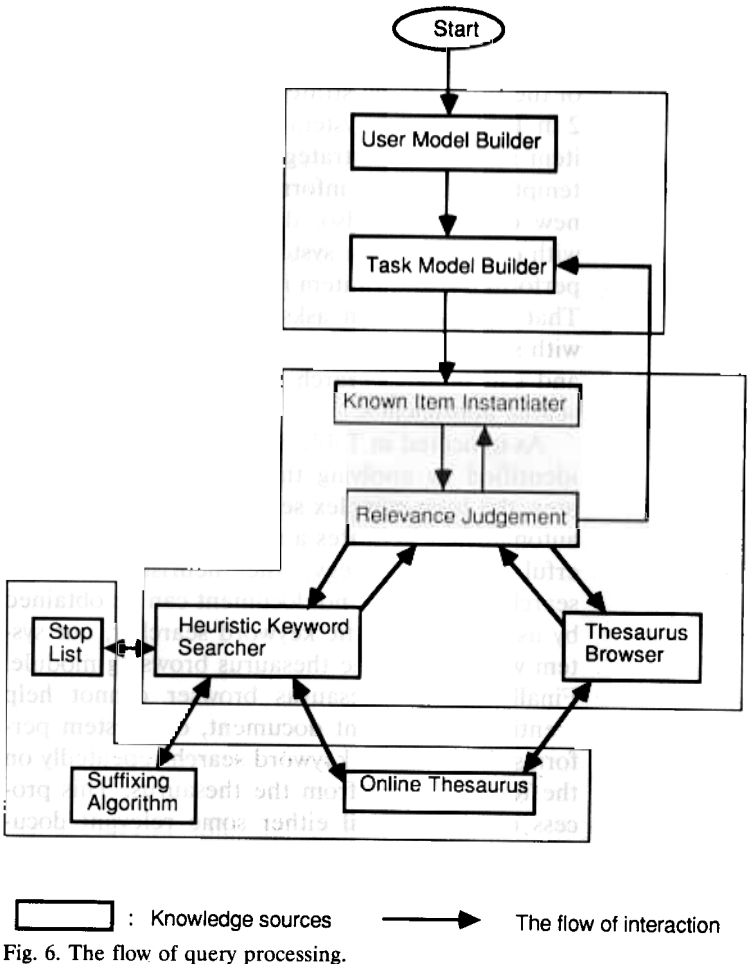

Fig. 6. The flow of query processing.

```
                *** SOLICIT TASK INFORMATION ***
  What is the purpose of your search? Is it for:
    1.  Class or course reading
    2.  A class project or paper
    3.  A research paper (for publication)
    4.  Thesis or dissertation
    5.  Other
  Enter a number, then press CARRIAGE RETURN
==> 5 (The searcher's selection.)
```

Fig. 7. Menus for eliciting task-related information.

ment module. The three search engines perform the actual search, based on the user/task model and the query model posted on the blackboard, and with the assistance of the online thesaurus, the suffixing algorithm, and the stop word list. The relevance judgement module solicits the searcher's evaluation of the index terms and citations generated by the three search engines. Searchers are requested to select the index terms and citations which they judge relevant. By repeatedly selecting and updating the relevant index terms and citations generated by the system (searchers are allowed to delete the index terms and citations which they no longer consider relevant), searchers' queries can be refined and their information needs can be satisfied. We summarize the flow of query processing as follows:

(1) *Formulating task requirements*: The system first solicits the long-term information of the searcher and the short-term task-related information (see the first two boxes in Fig. 6).

(2) *Invoking search engines*: Our system invokes the known item instantiater first. If nothing can be found or if the instantiater is not applicable, our system proceeds to the heuristic keyword searcher. If some initial citations have been generated from the system, the selection of the next search engine is determined by the control module we described. All three search engines are instantiated iteratively.

The suffixing algorithm and the stop word list support the heuristic keyword searcher. The online thesaurus, on the other hand, is used extensively by both the heuristic keyword searcher and the thesaurus browser.

(3) *Refining search via the searcher's relevance judgement*: As shown in Fig. 6, the relevance

```
                        *** KNOWN ITEM SEARCH ***
Please identify a few relevant citations known to you. Based on which
this system will be able to suggest similar citations to you.
Type in the number or three-letter code for the type of search you wish to do.

    1. AUT - Use when you know the name of the author, editor, cor
             author (conference, company name, etc).
    2. TIL - Use when you know the title of the book, journal, ser
    3. A-T - Use when you know both the author and the title.
    4. NUM - Use when you know the call number.
    5. NON - Use when you do not have any of the above information.
             The system will suggest other approach. -OR-

 TAS - display or update the task-related information you have supplied
 RES - display or update the search results you have derived
Enter number or code, then press CARRIAGE RETURN
==> 5
```

Fig. 8. Menus for the known item instantiater.

judgement function is at the center of the interaction. Any search results derived from the three search engines need to be evaluated by the searcher. Searchers' active participation in shaping and refining queries is an essential component in our design.

A prototype system, called Metacat, was developed in Franz Lisp, Opus 42, and run under SUN/3. FLAVORS, (which is part of Franz Lisp), was used to generate frame-based representations for the database records, the thesaurus entries, and the data on our blackboard-based system. Our system consists of roughly 6,000 lines of code, of which the majority are devoted to the three search engines. To provide searchers with a familiar environment, the system emulates the menu-driven interface available in most conventional online catalog systems. We present a sample user/Metacat interaction in this subsection.

In this example, the system first solicited the searcher's academic area and affiliation with the university. It then asked for task-related information including: the purpose of the search, the type of material the searcher wanted, the range of the years of publication, and the number of books the searcher desired. Fig. 7 shows the menu for eliciting the purpose of the search. In the example, the searcher was an MBA student majoring in MIS. She wanted 50 citations, from 1973 to 1989, for her task, which was to search for college-level mathematics-related textbooks.

After eliciting the user and task-related information, the interaction moved on to the solicitation of the query-related information. The known item instantiater was invoked first. The screen for the known item search is shown in Fig. 8. At this stage, the searcher can perform the known item search (Options 1 to 4 in Fig. 8), check the task-related information or the search results (Options TAS and RES), or choose Option 5 to move on to a heuristic keyword search. In this interaction, the searcher chose Option 5 because she did not know any citations in the area of her inquiry. This brought up the heuristic keyword searcher.

Searchers using the system can supply as many terms as they like for their queries. The actual search operations, however, are invisible to them —the heuristic keyword searcher performs the 10 search functions behind the scene without notifying the searchers. As in the interaction with the known item searcher, the subject has the option to check or correct the task-related information or the search results at this stage of search. In

```
                    *** DISPLAY MATCHED HEADINGS ***
                                                    Screen 1 of 2
  The system has derived the following subject headings:
Ref# Headings
   1 MATHEMATICS -- EXAMINATIONS, QUESTIONS, ETC
   2 MATHEMATICS -- STUDY AND TEACHING (SECONDARY) -- GREAT
   3 MATHEMATICS, ANCIENT
   4 MATHEMATICS
   5 MATHEMATICS -- 1961-
   6 MATHEMATICS -- STUDY AND TEACHING (ELEMENTARY)
   7 MATHEMATICS -- STUDY AND TEACHING (HIGHER)
   8 MATHEMATICS -- STUDY AND TEACHING -- MOLDAVIAN
   9 MATHEMATICS -- HANDBOOKS, MANUALS, ETC
  10 MATHEMATICS -- PROBLEMS, EXERCISES, ETC

 SEL - select relevant headings       FOR - move forward in this list
 BAC - move backward in this list     END - exit from this menu
 TAS - display or update the task-related information you have supplied
 RES - display or update the search results you have derived
 CON - choose Boolean operation or ranking options
Enter code, then press CARRIAGE RETURN
==> END (The system proceeds to find the citations.)
```

Fig. 9. The headings generated by the heuristic keyword searcher.

this example, the searcher used only one term, "MATHEMATICS," for her query. The system generated 15 subject headings with "MATHEMATICS" appearing in some position of the headings (SUBK with the original search term). (Even search terms like "MATH" or "MATHEMATICAL" would result in matches with MATHEMATICS-related subject headings.) The searcher selected three relevant subject headings (indicated by the "*" signs after the subject heading in Fig. 9). The system then displayed all citations matched by these subject headings. The searcher, however, did not find anything she considered relevant. Ending the session without selecting any of the system-generated citations caused the system to invoke the thesaurus browsing module automatically, using the three subject headings selected. By activating links and nodes in the knowledge base (see Fig. 10), the thesaurus browser generated 10 candidate subject headings in less than 10 seconds. These terms were either one link or two links away from the three selected subject headings. Since the starting three terms were closely related, these related terms were all in the same *concept group* and were displayed in decreasing order of relevance. The searcher selected five relevant terms, including: "ALGEBRA," "ARITHMETIC," "CALCULUS," "GEOMETRY," and "TRIGONOMETRY" (see the terms indicated by the "*" sign in Fig. 10). The system used these five terms to generate 12 citations, which were all selected by the searcher as relevant.

The whole search process lasted 19 minutes, with most of the time spent by the searcher in examining the system's suggestions and results. In this particular example, the thesaurus browser played an important role in helping the subject articulate her query and generating the relevant documents. Not knowing the system's underlying search engines, the subject was surprised by the system's capability to quickly suggest other semantically relevant, but syntactically different terms. She also had very little problem with the menu-selection interface provided by the system, which was similar to the operational online catalog system used in her university library.

The initial performance of our prototype system was promising. However, a more detailed and systematic testing is needed to examine the "intelligence" of such a system and its usefulness in assisting online information retrieval. We are also particularly concerned about users' being receptive to this type of system which performs a lot of "reasoning" for them (e.g., consulting the

```
                    *** DISPLAY MATCHED HEADINGS ***
                                                    Screen 1 o
  The system has derived the following subject headings:
Ref# Headings
   1 SCIENCE
   2 ALGEBRA
   3 ARITHMETIC
   4 CALCULUS
   5 DYNAMICS
   6 EQUATIONS
   7 GAME THEORY
   8 GEOMETRY
   9 NUMBERS, THEORY OF
  10 TRIGONOMETRY

 SEL - select relevant headings        FOR - move forward in this list
 BAC - move backward in this list      END - exit from this menu
 TAS - display or update the task-related information you have supplied
 RES - display or update the search results you have derived
 CON - choose Boolean operation or ranking options
Enter code, then press CARRIAGE RETURN
==> END
```

Fig. 10. The headings generated by the thesaurus browser.

thesaurus automatically and making suggestions). These issues involve potential resistance to using the system and remain to be studied and addressed.

5. Discussion

The research reported here attempted to establish a framework for knowledge-based information retrieval and to propose a design that can address significant issues within this framework. The specific linkages between our framework (see Fig. 1) and design (see Fig. 2) are discussed below:

Representing subject area and classification scheme knowledge: Searchers' general lack of classification scheme knowledge and their difficulty in articulating their subject-specific needs can be partly alleviated by the incorporation of a domain-specific online thesaurus. Such a thesaurus, when applied by the heuristic keyword searcher and the thesaurus browser, would allow most search terms to "dock" on the rich set of vocabulary provided by the system. Explicit cross-reference links and extensive searcher relevance feedback would help to foster a seamless human–computer collaboration during information retrieval.

Automatic system search: A searcher's lack of system knowledge (not knowing useful and powerful system commands to use) can be alleviated by the system's "automatic" heuristics keyword searcher and overall control module. When no results are derived by using the exact queries suggested by the searcher, the heuristic keyword searcher will take control and generate other possible syntactic (suffixing) and semantic variants (thesaurus lookup). This system-instantiated process, even though obtrusive, appears to be useful for searchers who are not familiar with the system's various search capabilities. At a higher level, the system's overall control module (see Table 1) will determine automatically what search engine to use next when no results have been generated by a previous search method.

This automated process is our attempt to incorporate explicit system knowledge in an "intelligent" system, instead of requiring searchers to be knowledgeable about the system's search capabilities. In a way, we are moving toward a more *declarative* search environment in which searchers specify *WHAT* they want (and the system decides HOW to do it) in contrast to the conventional *procedural* search environment in which searchers need to know exactly *HOW* to use the various search functionalities provided by an online catalog system. The effect of implementing such an knowledgeable and declarative online catalog system remains to be tested, however.

Simulating search strategies: Librarians' search strategies vary, but they are all dependent on knowledge of the classification scheme, system, and subject area. The three search engines incorporated in our system attempt to carry out some of the same procedures performed by reference librarians. The known item instantiater exploits the searcher's subject area knowledge. The heuristic keyword searcher utilizes the system's keyword searching, suffixing, and thesaurus term-switching capabilities. The thesaurus browser simulates a thesaurus consultation process. Even though our research did not exhaust all the search strategies used by human information specialists, it does shed light on the possibility of incorporating such human search strategies online.

User modeling: Our system's attempt to understand its users and their tasks (*user modeling* in Fig. 1) is demonstrated by the user model builder and the task model builder. In the current prototype we include only two dozen rules to help determine appropriate subject headings, suggest a reasonable number of citations, and perform a consistency check. These functionalities have not been fully evaluated due to the limited number of users we have tested. A more extensive acquisition of relevant user modeling rules from expert reference librarians and a full-scale evaluation of real-life searchers having different backgrounds and task requirements are needed. Developing these is planned as the next step of our research.

Several important features of our system are worth mentioning. First, the system falls into the category of "knowledge-based information retrieval systems." More specifically, it is a "heuristics-based" system that adopts various information specialists' and searchers' search strategies, thesaurus knowledge, and experts' user modeling heuristics in its search methods. Second, our system assists in the complete search process—from the formulation of the query and the selection of

search strategies to the actual execution of the search and the presentation and refinement of search results. It attempts to simulate the capabilities exhibited by human reference librarians and expert searchers in performing online information retrieval. Third, our system builds upon the functionalities and interfaces present in most conventional retrieval systems. Through the known item instantiater and the heuristic keyword searcher, the system has shown that, by using conventional search options "intelligently," it is possible to make inverted index based retrieval systems more effective and more powerful. Lastly, the heuristics-based branch-and-bound algorithm for thesaurus browsing is unique and useful. It makes spreading activation of nodes in a large semantic network computationally feasible.

Our research, which originated in the information science discipline (as shown in the *framework*) and was grounded on artificial intelligence techniques (as shown in the *design*), addresses problems common to document retrieval systems. We believe it has suggested promising directions for designing more useful and "intelligent" document retrieval systems.

References

[1] D. Appelt, The role of user modelling in language generation and communication planning. In: *User Modelling Panel*, Proceedings of the Ninth International Joint Conference on Artificial Intelligence, Los Angeles, CA, August 1985: 1298–1302.

[2] W.R. Ashby, *An Introduction to Cybernetics* (Methuen, London, 1973).

[3] M.J. Bates, Systems meets user: problems in matching subject search terms, *Information Processing and Management* 13(6) (1977) 367–368.

[4] M.J. Bates, Information search tactics, *Journal of the American Society for Information Science* 30(4) (1979) 205–214.

[5] M.J. Bates, Subject access in online catalog: a design model, *Journal of the American Society of Information Science* 37(6) (1986) 357–376.

[6] N.J. Belkin, R.D. Hennings and T. Seeger, Simulation of a distributed expert-based information provision mechanism, *Information Technology* 3(3) (1984) 122–141.

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

[8] N.J. Belkin, T. Seeger and G. Wersig, Distributed expert problem treatment as a model for information system analysis and design, *Journal of Information Science* 5 (1983) 153–167.

[9] L.M. Chan, *Library of Congress Subject Headings: Principles and Application*, 2nd ed. (Libraries Unlimited, Littleton, CO, 1986).

[10] H. Chen, An artificial intelligence approach to the design of online information retrieval systems (Unpublished PhD Thesis, Information Systems Department, New York University, 1989).

[11] H. Chen and V. Dhar, Reducing indeterminism in consultation: a cognitive model of user/librarian interaction. In: *Proceedings of the 6th National Conference on Artificial Intelligence* (AAAI-87), Seattle, WA, 13–17 July 1987: 285–289.

[12] H. Chen and V. Dhar, Online query refinement on information retrieval systems: a process model of searcher/system interactions. In: *Proceedings of the 13th Conference on Research and Development in Information Retrieval*, Brussels, 5–7 September 1990.

[13] H. Chen and V. Dhar, User misconceptions of online information retrieval systems, *International Journal of Man–Machine Studies* 32(6) (1990) 673–692.

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

[15] Y. Chiaramella and B. Defude, A prototype of an intelligent system for information retrieval: IOTA, *Information Processing and Management* 23(4) (1987) 285–303.

[16] P.A. Cochrane and K. Markey, Preparing for the use of classification in online cataloging systems and in online catalogs, *Information Technology and Libraries* 4(2) (1985) 91–111.

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

[18] W.B. Croft and R.H. Thompson, I^3R: a new approach to the design of document retrieval systems, *Journal of the American Society for Information Science* 38(6) (1987) 389–404.

[19] P.J. Daniels, The user modelling function of an intelligent interface for document retrieval systems. In: B.C. Brookes, ed. *Intelligent Information Systems for the Information Society* (North-Holland, Amsterdam, 1986) 162–176.

[20] D.L. Erman, F. Hayes-Roth, V.R. Lesser and D. Raj Reddy, The HEARSAY-II speech understanding system: integrating knowledge to resolve uncertainty, *ACM Computing Survey* 12 (1980) 213–253.

[21] E.A. Fox, Development of the CODER system: a testbed for artificial intelligence methods in information retrieval, *Information Processing and Management* 23(4) (1987) 341–366.

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

[23] L.M. Gomez and C.C. Lochbaum, People can retrieve more objects with enriched key-word vocabularies: but is there a human performance cost? In: B. Shackel, ed. *Human–Computer Interaction — Interact '84* (North-Holland, Amsterdam, 1984) 257–261.

[24] M.D. Good, J.A. Whiteside, D.R. Wixon and S.J. Jones,

Building a user-derived interface, *Communications of the ACM* 27(10) (1984) 1032–1043.

[25] R. Hjerppe, Project hypercatalog: visions and preliminary conceptions of an extended and enhanced catalog. In: *Proceedings of IRFIS, 6th*, Frascati, Italy, September 1985: 15–18.

[26] E. Horowitz and S. Sahni, *Fundamentals of Data Structures* (Computer Science Press, Woodland Hills, CA, 1976).

[27] B.L. Humphreys and D.A. Lindberg, Building the unified medical language system. In: *Proceedings of the Thirteenth Annual Symposium on Computer Applications in Medical Care*, Washington, DC, 5–8 November 1989.

[28] J. Jacoby and V. Slamecka, *Indexer Consistency under Minimal Conditions* (Documentation, Bethesda, MD, 1962).

[29] D.A. Lindberg and B.L. Humphreys, The UMLS knowledge sources: tools for building better user interface. In: *Proceedings of the Fourteenth Annual Symposium on Computer Applications in Medical Care*, Washington, DC, 4–7 November 1990.

[30] K. Markey, Levels of question formulation in negotiation of information need during the online presearch interview: a proposed model, *Information Processing and Management* 17(5) (1981) 215–225.

[31] K. Markey and P. Atherton, *Online Training and Practice Manual for ERIC Data Base Searchers* (ERIC Clearinghouse on Information Resources, Syracuse, NY, 1978).

[32] B.K. Martin and R. Rada, Building a relational data base for a physician document index, *Med. Inf.* 12(3) (July–September 1987) 187–201.

[33] A.T. McCray and W.T. Hole, The scope and structure of the first version of the UMLS semantic network. In: *Proceedings of the Fourteenth Annual Symposium on Computer Applications in Medical Care*, Washington, DC, 4–7 November 1990.

[34] I. Monarch and J.G. Carbonell, CoalSORT: a knowledge-based interface, *IEEE EXPERT* (Spring 1987) 39–53.

[35] H.P. Nii, Blackboard systems: blackboard application systems, blackboard systems from a knowledge engineering perspective, *AI Magazine* 7(3) (1986) 82–106.

[36] H.P. Nii, Blackboard systems: the blackboard model of problem solving and the evolution of blackboard architectures, *AI Magazine* 7(2) (1986) 38–53.

[37] P.L. Noerr and K.T. Bivins Noerr, Browse and navigate: an advance in database access method, *Information Processing and Management* 21(3) (1985) 205–213.

[38] R.C. Palmer, Search strategy development. In: *Library Science Text Series in Online Reference and Information Retrieval* (Libraries Unlimited, Littleton, CO, 1987) 74–94.

[39] S. Pollitt, Cansearch: an expert systems approach to document retrieval, *Information Processing and Management* 23(2) (1987) 119–138.

[40] R. Rada, H. Mili, E. Bicknell and M. Blettner, Development and application of a metric on semantic nets, *IEEE Transactions on Systems, Man, and Cybernetics* 19(1) (1989) 17–30.

[41] E. Rich, Building and exploiting user models. In: *International Joint Conference of Artificial Intelligence*, Tokyo, August 1979: 720–722.

[42] E. Rich, User modeling via stereotypes, *Cognitive Science* 3 (1979) 329–354.

[43] E. Rich, Users are individuals: individualizing user models, *International Journal of Man–Machine Studies* 18(3) (1983) 199–214.

[44] G. Salton, *Automatic Text Processing* (Addison-Wesley, Reading, MA, 1989).

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

[46] D. Sleeman, UMFE: a user modeling front-end subsystem, *International Journal of Man–Machine Studies* 23 (1985) 71–88.

[47] M.E. Stevens, *Automatic Indexing: a State-of-the-art Report* (US Government Printing Office, Washington, DC, 1965).

[48] W. Swartout, Explanation and the role of the user model: how much will it help? In: *User Modelling Panel*, Proceedings of the Ninth International Joint Conference on Artificial Intelligence, Los Angeles, CA, August 1984: 1298–1302.

[49] R. Tagliacozzo and M. Kochen, Information-seeking behavior of catalog users, *Information Storage and Retrieval* 6(5) (1970) 363–381.

[50] R.S. Taylor, The process of asking questions, *Am. Documen.* 13 (1962) 391–396.

[51] R.S. Taylor, Question-negotiation and information seeking in libraries, *College and Research Libraries* 29 (1968) 178–194.

[52] P.H. Winston, *Artificial Intelligence*, 2nd ed. (Addison-Wesley, Reading, MA, 1984).

[53] A.Y. Zissos and I.H. Witten, User modeling for a computer coach: a case study, *International Journal of Man–Machine Studies* 23 (1985) 729–750.