A Knowledge-Based Approach to the Design of Document-Based Retrieval Systems

Hsinchun Chen¹, MIS Department, University of Arizona
Vasant Dhar², IS Department, New York University

LOIS 92,
Cambridge, MASS

Abstract

This article presents a knowledge-based approach to the design of document-based retrieval systems. We conducted two empirical studies investigating the users' behavior using an online catalog. The studies revealed a range of knowledge elements which are necessary for performing a successful search. We proposed a semantic network based representation to capture these knowledge elements. The findings we derived from our empirical studies were used to construct a knowledge-based retrieval system. We performed a laboratory experiment to evaluate the search performance of our system. The experiment showed that our system out-performed a conventional retrieval system in recall and user satisfaction. The implications of our study to the design of document-based retrieval systems are also discussed in this article.

1 Introduction

Electronic information storage and retrieval systems in the form of online catalogs, online bibliographic databases, customized electronic newspapers, legal and financial databases, and videotex that can store huge amounts of data and allow access via terminals 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. While archival information sources are becoming increasingly computerized, access to such information is often difficult. This is due in large part to the indeterminism involved in the process by which documents are indexed, and the latitude users have in expressing a query.

In our research we studied computerized document-based retrieval systems. These systems store data, usually in text format, and retrieve it upon user request [15]. A typical document-based retrieval system consists of a database of documents, a classification scheme to index the documents, and an online system to access the documents.

In this article, we first present two empirical studies we conducted in Section 2. The findings of these studies are presented in Section 3. In Section 4, we report the design of a knowledge-based retrieval system we developed. We present the performance of this system in Section 5. The implications of our research to the design of document-based systems are discussed in the last section.

2 A Study of Document-Based Retrieval Systems

In this section, we present a study of a document-based system which we performed. The aim of the study was to investigate the knowledge components and the process models involved in the online document-based information retrieval process. The focus of this article, however, is on the knowledge components. Findings

¹hchen@mis.arizona.edu

²vdhar@vx1.gba.nyu.edu

© 1990 ACM 089791-358-2/90/0004/0281 \$1.50

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

pertaining to the process models can be found in [4] and [3]. Based on our findings, we designed a knowledge-based retrieval system. The performance of this system and its implications to other document-based systems will also be discussed in this paper.

Two field studies were conducted between October, 1986 and March, 1988 at New York University. Data collection techniques include think-aloud protocols, tape-recordings of the interactions, interviews, and questionnaires. The first study observed the interactions between users and reference librarians using an online retrieval system. The second study, on the other hand, studied the interactions between users and a retrieval system³ without the assistance of the information specialist. Based on these two studies, we were able to identify the knowledge components (or the lacks of knowledge components) that facilitate (or hinder) the retrieval process. The technique we used to analyze the collected data was *protocol analysis*, a qualitative analysis technique frequently used in the AI and cognitive psychology communities [8].

3 Knowledge Components: Subject Area, Classification Scheme and System Knowledge

in this section we report the findings derived from our empirical investigation. Specifically, we report the knowledge elements which are involved during information retrieval.

3.1 A Taxonomy of Knowledge Components

The logs and the think-aloud protocols revealed the knowledge elements which are essential for conducting a search. We identified three broad categories of knowledge elements: subject area knowledge, classification scheme knowledge, and system knowledge. Detailed discussions and examples are presented in [5].

3.1.1 Subject Area Knowledge

We have identified several subject area related knowledge elements.

A. Choosing Appropriate Terms

The system consists of *official terms* (index terms). Users on the other hand, use terms that they feel best express the *semantic content* of their queries. However, users may choose inappropriate terms (semantically inconect terms) to express their queries. Not surprisingly, users who are more familiar with the subject area (e.g. faculty members or Ph.D. students) tend to have less problems in generating appropriate terms.

B. Having an Estimate About the Volume of Relevant Work

Problems can also arise when a user does not have a good estimate about how much material exists in a subject area. Again, users who are more familiar to the subject area have less difficulties in estimating the amount of relevant work.

C. Expressing the Query at an Appropriate Level of Specificity

A common tendency among users is one of not expressing the query at an appropriate level of specificity. The use of a *broader entry first* strategy has been observed in other studies [1] [2]. In our investigation, over 70 percent of the subjects used terms that were more general than they should have been.

3.1.2 Classification Scheme Knowledge

There were three types of classification scheme related knowledge.

³Bobcat, NYU's online catalog. It lists over 600,000 catalog records. The system provides seven search options, namely, title search, author search, combination of title and author search, subject search, number search, keyword search, and Boolean search..these options are available in most online catalog systems.

A. Knowledge of Subject Headings

In contrast to situations where a user uses incorrect terms there are times when the user may in fact use correct terms, which nevertheless yield **no** matches **because** the system uses other synonyms for indexing. This type of problem stems from a lack of knowledge about the difference between official terms (index terms) and unofficial terms (non-index terms).

B. Knowing the LCSH Indexing Principles

A second problem was the lack of knowledge about the indexing principles of **LCSH**. There are three principles that were violated repeatedly. Based on the *specific entry principle*, subject headings assigned to a document are as specific as possible. However, users tended to think that a document classified under a certain heading should also be classified under broader headings. Secondly, in order to reduce redundancy, LCSH indexers are trained to use a term that indexes a whole book, not a portion of it. This is referred to as the *whole document indexing principle*. Again, users without a knowledge of the LCSH violated this principle. Lastly, most users were unaware of the *subdivisions* in LCSH. Standard subdivisions within subject headings are features like *topical* (which limit a concept term to a sub-topic), *period* (time), and *local* (e.g., geographic area). Without this knowledge, users tended to explore combinations of terms that for the most part were unproductive.

C. Knowledge about the LCSH Handbook

Only one out of the thirty subjects asked for or consulted the two volume LCSH Handbooks. When faced with difficulty in generating system-recognizable terms, even experienced users did not consult the handbooks.

Most users had limited classification scheme knowledge. The reference librarians, on the other hand, had a lot of classification scheme knowledge. Their searches are more effective and efficient due to their abilities to apply this type of knowledge.

3.1.3 System Knowledge

The last category of knowledge is about the system itself. System related knowledge were of two types: knowledge about the system's messages and display, and knowledge about the system's functionalities.

A. Understanding the System's Interface

Some system messages were misinterpreted by users due to a lack of precision in the system's language. At times, users confused about the menu selections displayed on the screen (Bobcat is a menu-driven system). They had incorrect interpretation of the various options.

B. Understanding the System's Functionalities

Knowledge of the various search options provided by the system is crucial to the success of a search session. However, there was a lack of clear understanding of the capabilities of each of the seven search options among our subjects. This misconception reflects a limited knowledge of the system's functionality and a low frequency of use of the system.

A second major problem relating to the system's capabilities was a lack of understanding of the system's match/search method. The system finds documents by matching alphabetically the terms supplied by the user. Six subjects exhibited a lack of understanding of this process.

The last type of problem resulted from a lack of knowledge about *levels* of menus in the system. Users had problems remembering their position during their interaction. Apart from limited knowledge about how to traverse the system menus, this type of wasteful search is probably the consequence of limited human short term memory. In the absence of an indication of the level of dialog from a system, users often tend to go back to the top level unnecessarily.

Figure 1: A portion of a semantic network

3.2 Representation of Knowledge Components

A semantic network structure is appropriate for representing subject area knowledge and classification scheme knowledge. The existing thesaurus (hardcopy), such as **LCSH** Handbooks, contains not only official terms (subject headings), but also *unofficial* terms which point to the official terms. The thesaurus can be viewed as a large semantic network of terms (concepts) where links are of two types: relations between unofficial and official terms, and set-superset relationships (i.e., **IS-A** links). Figure 1 shows a portion of the semantic network corresponding to the LCSH classification scheme. The terms and the cross referencing structures of the LCSH are incorporated.

Semantic network representation has been used by Cohen and Kjeldsen [6] and Shoval [14] for information retrieval. In contrast to their manual creation of the semantic network knowledge base, we proposed an automatic process of generating this semantic network knowledge base from an existing thesaurus (LCSH Handbook). We also developed a heuristic-based online thesaurus browser to assist query refinement. Details of this knowledge base and browser are presented in the next section.

4 A Knowledge-Based Design

In this section we present a knowledge-based design for online catalogs based on the findings we derived from our empirical studies. This system adopted the blackboard architecture. We present the system architecture in this section. In particular, we will discuss in detail two important knowledge-based components in our system: the online thesaurus and the thesaurus browser.

The first blackboard system was the **HEARSAY-II** speech understanding system [9] that evolved between 1971 and 1976. Subsequently, many systems have been built that have similar organization. Typically a blackboard structure consists of three components: the blackboard data structure, the knowledge sources, and the control module [12].

Figure 2: The blackboard architecture of Metacat

4.1 The Blackboard

In our system (Metacat) there are five types of data posted on the blackboard (as shown on the left of Figure 2). All of them are represented in *frame*. Each frame consists of a few attributes (slots). The five data types are organized as a hierarchy in the blackboard to depict their closeness to the user's information need. **At** the highest level, a user model is built, which captures the long term characteristics of the user, such as the user's educational background, the user's educational level, etc. At the second level, a task model is constructed to represent the user's task related information, such as the type of material, the recency of the publications, and the amount of relevant citations the user desires. At the third level, the system creates a model of the actual query, which includes all the query-related inputs supplied by the user. These include: dtles of books, author's names, call numbers, and search terms. At the fourth level, index terms are generated by the system to capture the user's query. **At** the fifth and last level, citations that are derived from the index terms are posted on the blackboard. The information on the blackboard is constantly revised by the various knowledge sources.

4.2 The Knowledge Sources

Our system consists of eight knowledge sources (as shown on the right of Figure 2). Each knowledge source is capable of performing specific function.

Firstly, the user model builder creates a long term model of the user by soliciting the user's id. number, educational level, and educational field. Based on this information, the user model builder uses a few heuristics to infer the user's familiarity levels in **the** subject area, the classification scheme, **and** the system.

Secondly, the **task** model builder solicits the user's information concerning the type of material the user wants, the purpose of the search, the number of citations expected, **and** the recency of the publications desired by the user. It then applies its production rules to determine the amount of citations the system should bok for and the characteristics of the subject headings.

Thirdly, our system includes a stemming algorithm. This algorithm first identifies the root word of the input word arid then subsequently generates a list of legal suffixed words from the root word.

Fourthly, words which do not bear meanings are included in a stopwords list in our system. This list filters the searcher's input.

Our system consists of three search engines. The first search engine is the known **item** instantiator (the fifth knowledge source). Users identify some relevant citations through the use of the conventional known item search options. The system then uses the subject headings assigned to these retrieved citations to perform a controlled subject search (SUB). This strategy can be applied continuously in a "chain reaction" way. 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 continuously, we can identify a set of relevant subject headings and citations.

The heuristic keyword searcher (the sixth knowledge source) is another search engine which utilizes the existing search options in a new way. While the known item instantiator exploits the known item search options, the heuristic keyword searcher uses the non known item search options, in particular, the controlled subject search (SUB) and the title search (TIL). This heuristic keyword searcher consists of ten different search functions. Each function uses different search option, such as perform SUB (subject search) using the user's search term, perform **SUBK** (subject keyword search) using the suffixed search term, perform TILK (title keyword search) using suffixed search term, etc. Each search function has a different level of *credibility* indicating its likelihood to generate 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 credible search function. This process continues until either the system finds some index terms by using a certain search function or until the system exhausts all ten search functions. In the following subsections, we discuss the last two knowledge sources in our system in detail.

4.2.1 Online Thesaurus

Our system contains an online thesaurus corresponding to a portion of the **LCSH** Handbook. It consists of the terms and the cross references. This online thesaurus is represented as a semantic network where nodes are the terms (the official terms and the unofficial terms) and the links are the relationships between terms. We schematized this network in Figure 1.

We derived our semantic network based online thesaurus by extracting a portion of the computer readable form of the LCSH Handbook (with the assistance of OCLC⁴). Our online thesaurus consists of nearly 3,500 terms, (both official terms and unofficial terms) in the areas of mathematics and computer science (the areas we used for the system evaluation). Each term has between a couple of and a few hundred relevant terms associated via the cross referencing structure of the thesaurus. This online thesaurus represents the domain knowledge. It was constructed in FLAVORS, an object-oriented language. Since the complete LCSH is in an online format, the process of generating a semantic network representation is easy. With the proper heuristics, the online thesaurus can be extremely useful in assisting the users in refining their queries.

4.2.2 Thesaurus Browser

The last search engine of our system is the thesaurus browser. During a search, our system first invokes the **known** item instantiator and the heuristic keyword searcher. If the user is not satisfied with the results derived from these two search engines, the system will then instantiate the thesaurus browser.

The index terms generated from the above two search engines can match with some nodes in our online thesaurus. These nodes are taken as the **source** nodes by our thesaurus browser. There are links associated with these nodes. Our system applies a heuristic spreading activation process (activating the links to the source nodes) on the thesaurus to generate relevant terms. In particular, we developed a few heuristics to guide the activation process. They are presented as follows:

⁴Online Computer Library Center, Inc., Dublin, Ohio

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-net are generally more specific (in content) than nodes (terms) which have more neighbors. Since the users have a tendency of stating their information needs broader than they should (described earlier), our system applies a heuristic which expands nodes with fewer neighbors first (the more specific terms).

2. The Specific Links First Heuristic: Links associated with the official terms are of three types: NT, RT, and BT (the reverse of NT). Our system adopts a heuristic for expanding the links in the order of: NT, RT, and BT. That is, our system would search the NT links before it searches the RT links, and before it searches the BT links. This heuristic encourages searches on more specific links (which will eventually lead to more specific terms).
3. The Shorter Distance **First** Heuristic: This heuristic is related to the distance of a node to the source nodes. During the activation process, our system would expand the 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 terms than terms which are closer to the source nodes. Therefore, they should be expanded only after the more relevant terms (closer nodes) are expanded.
4. The **Two** Level Expansion Heuristic: The number of links between two nodes (terms) in a semantic network determines the semantic proximity of the terms. In order to find only terms which are closely relevant to the source nodes, our system expands the source nodes by two levels. That is, we only activate nodes that are two links away from the source nodes. (Because each node in the network may have a few dozen to a few hundred links, two level expansion still requires a lot of computation.) This two level expansion heuristic was derived from the query refinement characteristic of the users (they rarely expanded the specificity level of their search terms for more than two levels) and an analysis of the cross referencing structure of LCSH. This heuristic ensures that the system only finds terms that are semantically close to the source nodes (terms).

The above four heuristics, which consider the specificity of the nodes, the specificity of the links, the distance of the nodes, and the expansion level, are used to direct our system's spreading activation effort. We represent the problem as a search task. We assign costs to each expanded paths based on the nodes they visit, the types of links they traverse, and the number of links in the paths. We develop a branch-and-bound algorithm to guide the search. This algorithm computes costs for each partial paths and expands the least-cost path (after the expansion, new costs are assigned and the expansion process repeats again). The algorithm is able to alleviate the computational explosion problem that frequently occurred in the semantic-net spreading activation process. Details of this algorithm appear in [3].

After expanding each source node by two levels and after following our search heuristics, our system classifies the expanded paths into different concept groups. Paths that are linked together 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. For queries which involve one concept (topic), our system will generate one concept group. For queries which involve multiple concepts (topics), our system is able to generate multiple concept groups. This concept grouping process had been observed among human information specialists (reference librarians often attempt to identify the various topics patrons try to address during a consultation session). New terms (terms which are different from the source terms) found in each concept group can become good candidate terms for the users' 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 users. A session of user relevance feedback then follows.

In summary, this thesaurus browsing process generates relevant terms by activating the links in the network based online thesaurus. This process has been made possible via a heuristic-based branch-and-bound algorithm. The thesaurus browser, along with the known item instantiator, and the heuristic keyword are applied iteratively in our system.

4.3 The Control Module

The control module of our system is data-driven. Based on the partial search results posted on the blackboard (the index terms and the citations generated during the search) and the search engines that are applicable, our system chooses the next action to apply automatically.

Our system was developed in Franz Lisp, Opus 42, and run on the **SUN.FLAVORS**, an object oriented language (which is also part of the Franz Lisp), was used to represent the database records, the online thesaurus, and the data on our blackboard-based system. The knowledge sources are represented using Lisp.

5 System Evaluation

We conducted a system evaluation in the summer of 1989. Forty subjects participated in this experiment, ranging from undergraduate students to faculty members. These subjects were randomly assigned to one of the two groups: the control group (using Bobcat), and the test group (using Metacat). Each group consisted of twenty subjects. Each subject was asked to perform two same pre-determined tasks. Both tasks were complex. The first task was taken as the training task. The second task (the task used for evaluation) was to find textbooks that address basic computer science-related concepts for an introductory course. We also asked three experts to select a list of relevant books from the database for each task. Based on the citations generated by the subjects and the experts and other performance related information we collected from the questionnaire, we were able to evaluate our system's performance. Specifically, we developed four hypotheses for the system evaluation. These hypotheses are:

1. There is no difference between the search performance as judged by Bobcat users and Metacat users.
2. There is no difference between the amount of time Bobcat users spend in performing searches and the amount of time Metacat users spend.
3. There is no difference in recall between searches on Bobcat and searches on Metacat.
4. There is no difference in precision between searches on Bobcat and searches on Metacat.

We used the Student's t-test for analysis. The results are shown in Table 1. We summarize the results from our hypothesis testing as follows:

1. Metacat users had higher satisfaction level than the Bobcat users (at the significance level of 0.043). On a scale of 4, the Metacat users' average satisfaction level was 3.300; while the Bobcat users' average satisfaction level was 2.800. Most Metacat subjects were happy with the results generated from their searches and the processes involved in deriving these results.
2. There was no significant difference between the amount of time Bobcat users spent in performing searches and the amount of time Metacat users spent. While users of Bobcat spent time making their decisions during the search, our system spent time on the actual computation.

	Bobcat	Metacat	Level of Sig.
Satisfaction	2.800	3.300	0.043
Search time	810 (secs)	935	0.310
Recall	0.098	0.174	0.038
Precision	0.549	0.555	0.940

Table 1: Results from t-test

3. The average recall values of the searches on Metacat were significantly higher than the average recall values of the searches on Bobcat (**78%** better in recall). Our system was able to suggest more relevant citations to the users than the traditional system.
4. There was no significant difference in precision between the two groups. We postulate that precision is related to the users' subject area knowledge. Due to the random assignment of subjects to each group, it is not surprising to see no difference in precision.

6 Discussions

In our research we identified three types of knowledge which are necessary for performing a successful search. These include: the subject area knowledge, the classification scheme knowledge, and the system knowledge. In order to make the document-based retrieval systems "intelligent" and active in assisting search, document-based systems need to incorporate a knowledge base. The semantic network based online thesaurus we proposed is appropriate for this purpose. Currently, **LCSH Handbook** includes most of the academic areas and it is available in the computer-readable form. The generation of an initial knowledge base is an automatic process. For a given application area (lets say, finance or marketing), we can derive a knowledge base by extracting the relevant terms and relationships between these terms from the LCSH Handbook. A system designer can also expand this knowledge base by incorporating other thesauri or other domain-related terms. This knowledge base along with a heuristic-based browser (like the one we developed) can assist users in articulating their queries and revealing their information needs.

Lastly, the blackboard architecture we proposed is suitable for constructing "intelligent" document-based systems. Similar blackboard architectures have also been adopted by Croft and Thompson [7] and Fox [10]. The design that we proposed, however, exhibited several unique characteristics. Firstly, it adopted a uniform representation for the objects involved in information retrieval. They were all represented in *frame*. Secondly, with the incorporation of the domain knowledge, our system provided a *semantics-based* search assistance. This presents an alternative to the *keyword-based* search assistance (in the conventional inverted index-based retrieval systems) and the *statistics-based* search assistance (in the statistics-based retrieval systems) [13]. Thirdly, in contrast to the *term-oriented* search performed by the conventional retrieval systems (search terms used by the searcher are taken as independent), our system performed a *task-oriented* search. All the information and results generated during the search were used to represent a user's *task*. It helps the retrieval system identify the users' information needs.

Our research, which originated from the information science discipline and which was grounded on the

artificial intelligence research methodology, addresses problems common to document-based retrieval systems. We believe our research has contributed to the **understanding** of document-based information retrieval process and to the **design** of "intelligent" and useful document-based retrieval systems.

References

- [1] Marcia J. Bates. Systems meets user: Problems in matching subject search terms. *Information Processing and Management*, 13(6):367-368, June 1977.
- [2] N.J. Belkin, R. N. Oddy, and H.M. Brooks. Ask for information retrieval: Part I. background and theory. *Journal of Documentation*, 38(2):61-71, June 1982.
- [3] Hsinchun Chen. *An Artificial Intelligence Approach to the Design of Online Information Retrieval Systems*. New York University, Unpublished Ph.D. Thesis, 1989.
- [4] Hsinchun Chen and Vasant Dhar. Reducing indeterminism in consultation: a cognitive model of user/librarian interaction. In *Proceedings of the Sixth National Conference on Artificial Intelligence (AAAI-87)*, 1987.
- [5] Hsinchun Chen and Vasant Dhar. User misconceptions of online information retrieval systems. *International Journal of Man-Machine Studies*, Forthcoming, 1990.
- [6] Paul R. Cohen and Rick Kjeldsen. Information retrieval by constrained spreading activation in semantic networks. *Information Processing and Management*, 23(4):255-268, 1987.
- [7] 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):389-404, 1987.
- [8] K. Anders Ericsson and Herbert A. Simon. *Protocol analysis: verbal report as data*. The MIT Press, Cambridge, Massachusetts, 1984.
- [9] 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*, 12:213-253, 1980.
- [10] Edward A. Fox. Development of the CODER system: A testbed for artificial intelligence methods in information retrieval. *Information Processing and Management*, 23(4):341-366, 1987.
- [11] J. Minker. Information storage and retrieval - a survey and functional description. *ACM/SIGIR*, Fall, 1977.
- [12] H. Penny Nii. Blackboard systems: The blackboard model of problem solving and the evolution of blackboard architectures. *AI Magazine*, pages 38-53, Summer 1986.
- [13] G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983.
- [14] Peretz Shoval. Comparison of decision support strategies in expert consultation systems. *International Journal of Man-Machine Studies*, 24:475-487, 1986.
- [15] E. B. Swanson and M. J. Culnan. Document-based systems for management planning and control: sification, survey, and assessment. *The Management Information Systems Quarterly*, 2(6):31-46, December 1978.