Evaluation of an Expert System for Searching in Full Text

by

Susan Gauch

Department of Computer Science
Wellesley College
Wellesley, MA 02181

ABSTRACT

This paper presents a prototype expert system which provides online search assistance. The expert system automatically reformulates queries, using an online thesaurus as the source of domain knowledge, and a knowledge base of domain-independent search tactics. The expert system works with a full-text database which requires no syntactic or semantic pre-processing. In addition, the expert system ranks the retrieved passages in decreasing order of probable relevance.

Users' search performance using the expert system was compared with their search performance on their own, and their search performance using the online thesaurus. The following conclusions were reached: 1) The expert system significantly reduced the number of queries necessary to find relevant passages compared with the user searching alone or with the thesaurus. 2) The expert system produced marginally significant improvements in precision compared with the user searching on their own. There was no significant difference in the recall achieved by the three system configurations. 3) Overall, the expert system ranked relevant passages above irrelevant passages.

1 INTRODUCTION

1.1 Online Search Difficulty

Information systems are undergoing a technological revolution. Massive quantities of online text are being produced using optical character recognition hardware, word processors, and computer publishing software. These large full-text databases, or *textbases*, are being stored and distributed

(C) 1990 ACM 0-89791-408-2 90 0009 255 $1.50

on optical storage media. End-users are searching online databases themselves, with the use of personal workstations and modems. Soon, the inability of end-users to search effectively will be the main roadblock to the wide-spread use of online textbases.

Christine Borgman [Borgman, 1986] identifies two types of knowledge necessary to search: knowledge of the mechanical aspects of searching, and knowledge of the conceptual aspects. She concludes that whereas system mechanics are rarely a problem for any but very inexperienced and infrequent users, even experienced searchers have significant problems with search strategy and output performance. Similarly, Carol Fenichel [Fenichel, 1981] finds that even experienced searchers could improve their search results by using more system interaction to iterate their search.

Studies of inexperienced searchers find even more problems with search strategy. In one study [Borgman, 1987], a quarter of the subjects were unable to pass a benchmark test of minimum searching skill. Another experiment [Oldroyd, 1984] found that novices find some relevant documents easily, but they fail to achieve high recall and are unable to reformulate queries well.

David Blair [Blair and Maron, 1985] paints an even bleaker picture for searching full-text databases. Lawyers searching a legal database achieved only 20% recall, although they were attempting to do a high recall search. The factors, as identified by the authors, leading to this poor performance were poor searching technique, stopping the query iteration too soon, and the inability to search on inter-document relationships.

1.2 Research Overview

My goal is to demonstrate that an expert system can improve a novice searcher's retrieval from full-text databases. To this end, I have developed a expert system which automatically reformulates user queries and ranks the retrieved passages. The expert system incorporates a knowledge base of domain-independent search tactics, ranking rules, and has access to an online thesaurus.

1.3 Related Research

Bibliographic Expert Systems

Steven Pollitt [Pollitt, 1984] built an expert system to search the MEDLINE medical database for cancer literature. The knowledge base is tailored for the specific database queried. Peretz Shoval [Shoval, 1985] has developed an expert system which uses the users' initial search terms to identify nodes in a semantic network of search terms. The links from these nodes are used to identify new, potentially relevant, search terms. These new terms are given strength ratings and suggested to the user as possible alternative search terms. IR-NLI II [Brajnik et al, 1988] incorporates user modelling into a domain-independent bibliographic retrieval expert system. IOTA [Chiaramella and Defude, 1987] is a bibliographic expert system which incorporates a natural language interface. PLEXUS [Vickery and Brooks, 1987] is an expert system designed to help novice users find information about gardening. Natural language queries are accepted, and information is extracted to fill in frames.

Each of latter four systems uses an online classification of terms, similar to a thesaurus, as the source of domain knowledge. However, only PLEXUS and IOTA incorporate strategies to automatically reformulate queries, although that is not the main focus of those systems.

Full-Text Expert Systems

Fewer projects are aimed at providing intelligent assistance for full-text searching. One such system is RUBRIC [Tong et al, 1987], which has the user describe his query in terms of rules. These rules describe the domain knowledge for the system as a hierarchy of topics and subtopics. I^3R [Croft and Thompson, 1987] also requires the user to provide the appropriate domain knowledge. The query process is managed as a dialogue between the user and the system during which the user is asked to supply a semantic network, similar to a thesaurus, that describes the relationships among the concepts in his query. A full-text system that incorporates query reformulation assistance is under development at OCLC [Teskey, 1987]. The emphasis to date has been on provision of an intelligent online help function, but a few basic reformulation strategies are provided. The CODER system [Fox, 1988] incorporates natural language processing with expert systems techniques to produce a testbed for evaluating advanced information retrieval techniques. The expert system is used to identify the structure within electronic mail messages, and semantic relationships between messages.

Searching Studies

The most thorough catalogue of search tactics was compiled by Marcia Bates [Bates, 1979]. She outlined 29 search tactics in four areas: monitoring, file structure, search formulation, and term manipulation. Philip Smith [Smith, P. J. et al, 1989] conducted a similar study as the first step to building an online search intermediary for searching the environmental literature of Chemical Abstracts. By analysing the discourses and actions of 17 users and search intermediaries, he compiled a list of 19 search tactics. P. W. Williams [Williams, 1984] developed a model of all possible search situations and all possible responses, to be used as the basis of an expert system's knowledge base.

2 SYSTEM DESCRIPTION

The prototype search assistant system was implemented on a Sun3 workstation. It consists of five modules:

1) MICROARRAS [Smith et al, 1987], which serves as the full-text search and retrieval engine
2) a full-text database of over 188,000 words, containing a draft of "Computer Architecture, Volume 1 - Design Decisions" [Blaauw and Brooks, 1986]

3) a hierarchical thesaurus of approximately 7424 words specific to the textbase's domain
4) an expert system of 85 OPS83 rules and over 5,000 lines of C code, which interprets the user's queries, controls the search process, analyses the retrieved text, and ranks the search results
5) a user interface, which accepts the user's queries, presents requests for information from the expert system, and displays the search results.

The search process consists of a dialogue between the user and the expert system. The user enters the initial Boolean query and the number of passages he would like to retrieve. The query is parsed and translated into a request for information from MICROARRAS. MICROARRAS retrieves text passages from the full-text database and informs the expert system of the number of passages that satisfy the request. The expert system compares the number retrieved with the target number to decide how to reformulate the query.

To expand a search query, the expert system may use three different strategies, alone or in combination. First, it can expand individual search terms to the sets of words using the thesaurus. Words with the same stem, synonyms, broader, narrower, and similar words can be added iteratively. Second, it can relax contextual constraints. Since MICROARRAS provides considerable generality in terms of segmental contexts, search expressions may contain contextual parameters expressed in terms of any number of words, sentences, paragraphs, etc. to either the right or left of any term in the search expression. Thus, the expert system can increase the default number or type of such units to generate more potential hits. Finally, it can change the Boolean operators, making the query less restrictive by replacing ANDs with ORs or removing ANDNOTs.

To restrict a search, the expert system uses the same strategies as those described above, but in reverse. That is, it may add sets of search terms to those terms to be excluded from the retrieval passages, contract contexts, and replace ORs with ANDs. Changing the Boolean operators in this way will reduce the number of passages retrieved, in general, however, it is only likely to be useful when the user has used the incorrect Boolean operator in the original query.

Once an appropriate number of passages is identified, the expert system attempts to rank the passages in terms of probable relevance. It does this by performing a rudimentary content analysis on the passages retrieved by MICROARRAS and computing a relevance index for each. The relevance index for each passage is a function of the number of search terms actually found in that passage, the number of distinct types for each (for terms that are sets), and the number of different thesaural categories represented. Query structure, distance between search terms, and frequency of the search terms in the textbase as a whole are also taken into consideration. The retrieved passages are then sorted by their relevance indices and presented to the user in order of probable interest.

A major advantage of this architecture is the separation of strategic knowledge, contained in the knowledge base for the expert system, from domain knowledge, contained in the thesaurus. Now that the search strategy rules have been developed and tested with the existing textbase, the expert system can be tested with other content domains by simply providing a suitable thesaurus for the new textbase.

For a more complete description of the system's architecture and search strategies, see [Gauch and Smith J. B., 1989a]. In addition, [Gauch, and Smith, J. B., 1989b] contains a description of the implementation of the search strategies as rules in a knowledge base, and [Gauch, 1989] contains a complete description of the entire research project.

3 EVALUATION

Evaluating an interactive system is difficult. Jean Tague [Tague and Schultz, 1988] has defined a framework for evaluating information retrieval system interfaces. She identified three ways to measure the information retrieval system: informativeness, time, and user friendliness. Informativeness is measured by retrieval output (search effectiveness) and retrieval order (ranking). The search efficiency of the system is related to Tague's time factor. Finally, the user friendliness of the system can be evaluated by a post-search questionnaire.

My primary goal is to demonstrate that using an expert system to reformulate queries can improve search performance for novice searchers. Ideally, both their effectiveness and efficiency would be improved. The second, less important, goal is to show that the expert system can rank the retrieved passages in decreasing order of relevance.

To evaluate the expert system, subjects attempted to find relevant passages in response to high-level questions. They queried MICROARRAS with three interfaces with different capabilities: an interface whose only function was to accept contextual Boolean queries and display search results; a similar interface which also allowed the user to explore the online thesaurus; and a third which incorporated the searching expert system. Each subject's search performance with the three interfaces was monitored and compared.

3.1 Hypotheses

Hypothesis 1: The expert system improves the search effectiveness for a novice searcher.

Hypothesis 2: The expert system improves the search efficiency for a novice searcher.

Hypothesis 3: The expert system can rank the passages retrieved by the search in decreasing order of relevance.

The effectiveness of the retrieval output is evaluated by looking at recall (the number of relevant items found / the total number of relevant items in the database) and precision (the number of relevant items retrieved / the number of items retrieved). Two estimates of the number of relevant items retrieved are examined: the number of passages the users mark as relevant and the number of passages retrieved from the set of passages deemed relevant by the author.

The efficiency of the systems is measured by the number of Boolean queries the subjects entered for each of several high-level questions, and by the amount of time they spent searching for relevant passages for each question.

The ranking algorithm was evaluated by comparing the order of appearance of relevant passages after they have been ranked with a random order of appearance.

3.2 Method

Subjects

Twelve computer science graduate students participated as subjects in the study. All subjects were knowledgeable in the use of computers, but unfamiliar with online searching. Thus, they were representative of the anticipated users of future information retrieval systems.

Apparatus

Information Retrieval Systems

The *user-alone* configuration consisted of a Sun 3 running MICROARRAS and a rudimentary expert system. This expert system performed only the system control function, and did no query reformulation or ranking of retrieved passages. The user was prompted for a contextual Boolean query, this query was sent to MICROARRAS, and the number of passages retrieved was reported back to the user. The user could display the passages retrieved, if there were fewer than 25, or try another query.

The *user-thesaurus* version consisted of a Sun 3 with one window running MICROARRAS, as in the user-alone system, and a second window running a thesaurus access function. In the thesaurus window the user had access to all the thesaurus information available to the expert system. He could find out the stemname for a specific word's stemgroup. For any stemname, he could ask for the stemnames of the corresponding synonym, parent, sibling, or child stemgroups. These stemnames could be used in the user's query to MICROARRAS.

In the *user-expert system* version the user did not have access to the online thesaurus. Context and the addition of stemgroups were controlled by the expert system. Thus, the user entered a Boolean query and a target number of passages and the expert system reformulated the user's query to attempt to get close to the target number. The user was prompted to filter search terms found in the thesaurus, and to continue or abandon the current reformulation.

To keep the response time approximately the same as for the other two configurations it was necessary to run MICROARRAS remotely on the Sun 4 file server containing the textbase. The user worked with one window on a Sun 3 which ran the full version of the query reformulation expert system. The expert system communicated with MICROARRAS over the network. This setup was approximately twice as fast as when MICROARRAS was run on the user's Sun 3. This speed up was necessary, not because the expert system code itself was slow, but rather because the expert system tended to form very long queries involving many MICROARRAS categories, and MICROARRAS slows down linearly with the number of search terms in a query.

Questions

Three sets of five questions were devised. Each set contained one training question and four questions on which the subjects were monitored. The questions covered material ranging over the whole textbase. The number of relevant passages found by the author (see Definitions) follows each monitored question.

Query Set A

Practice:

What are some sources of error in floating point arithmetic?

Monitored:

1) How is computer architecture distinguished from the other computer design domains? (16)
2) What are some upward pressures on the level of a machine language? (16)
3) Fixed length multiplication produces a double length result. How have different machines handled this? (14)
4) How are interrupts handled? Do not consider techniques to disable them. (23)

Query Set B

Practice:

I/O devices have moving parts. What is the effect of this motion on the architecture of computers?

Monitored:

1) What are some design principles that lead to clean architectures? Do not consider the economic advantages of a quality design. (14)
2) What techniques have been used to reduce bit traffic? (10)
3) How are control structures implemented? (13)
4) What role does buffering play in I/O transfers? (22)

Query Set C

Practice:

Fragmentation of memory is one problem of using a segmentation scheme. How is paging used to fix this?

Monitored:

1) Discuss the two fundamentally different ways to formally specify an architecture. (19)
2) What are the effects of having two zeros, as in the sign magnitude representation of fixed point numbers? (7)
3) What is done to save state upon a procedure call? (15)
4) Besides I/O, where is concurrency practiced in the implementation? (16)

Procedure

Subjects were asked to try to find on the order of ten relevant passages from the textbase in response to the questions they would be given. They were informed that they might not always be able to find that many, and they were allowed to stop working on a query whenever they were satisfied that they had found as much as they could. The target number of ten was chosen because it was large enough to require a high recall search, yet small enough that the users would not become tired reading passages. For similar reasons, Carlo Vernimb [Vernimb, 1977] also used a target number of ten when developing an automatic query reformulation system for document retrieval.

Each subject worked with each of the three systems, in turn. This was done to compensate for the large individual differences found in searching ability [Borgman, 1987]. To compensate for learning during the experiment, the order of presentation of the three systems was counterbalanced among subjects. The subjects received a training session with each system before they began their monitored searches. When they had completed all three sessions, they were asked to fill out the questionnaire stating their preferences and opinions.

Data Collection

Raw Data

Data was collected in a trace file while the subjects worked with the system. Each communication from the subject to the retrieval system, and vice versa, was stored with a time stamp. Thus, timing information was collected along with the history of queries entered by the subject and the search results. When the subject chose to display the retrieved passages, those passages and the subject's relevance judgement of them were also stored.

Several parameters were chosen from the trace file to represent each subject's sessions. Measurements were taken on time, number of queries, and number of relevant passages.
Before the variables to be compared are described, I will provide a few definitions.

Definitions

A *unique query* was any error-free query entered by a subject. If a subject entered a query which contained a typographic or logical error, and he indicated that he noticed the error by aborting the search and re-entering a corrected version, then the erroneous query was not considered a unique query. However, if the subject gave no indication that he was aware of the error, but instead moved on to a different query altogether, then the erroneous query was considered unique.

The *relevance weight* of a passage is the relevance number assigned to the passage by the subject. A *very relevant (user)* passage is one assigned a relevance weight of two. A *somewhat relevant (user)* passage has a relevance weight of one. A *relevant passage (user)* is one that is either very relevant or somewhat relevant, as judged by the user. An *irrelevant passage (user)* is a passage given a relevance number of zero.

It is necessary to have an estimate of the total number of relevant passages available for each question, in order to calculate recall. This estimate was calculated by forming the union, for each question, of the set of passages judged very relevant by any subject. Passages in this set judged irrelevant by the author were removed. The remaining passages form the *absolute retrieval set* and are called the *relevant passages*. It was necessary to remove some passages marked very relevant by a subject because, perhaps due to a misinterpretation fo the question or a misunderstanding of the passage, some subjects gave a relevance weight of two to irrelevant or marginally relevant passages. This tendency to oversestimate the relevance of passages may also be because, in some cases, subjects were unable to find the truly relevant passages, and thought that they had retrieved the best passages available when in fact they had not.

A *successful retrieval set* is a retrieval set containing at least five relevant passages. Since the subjects were attempting to find ten relevant passages, a successful retrieval set contains at least half the number for which they were looking. The textbase contained approximately the same number of relevant passages for each question, allowing the target number and size of the successful retrieval set to be held constant.

The *final retrieval set* was chosen as the last successful retrieval set. If a subject never retrieved a successful retrieval set for a given question, the retrieval set with the highest number of relevant passages, as judged by the subject, was chosen.

Variables

Total time per question is calculated from the entry of the subject's first query for the question until after the display, or decision not to display, of the final set of retrieved passages.

Number of queries per question is determined by counting the number of unique queries the subject entered for a given question.

Number of relevant passages (user) found per question is determined by counting the number of user indicated relevant passages in the final retrieval set for the question.

User precision is calculated for the final retrieval set using the standard formula of:

number of relevant passages (user) retrieved / number of passages retrieved

Number of relevant passages found per question is determined by counting the number of passages in the final retrieval set for the question that are members of the absolute retrieval set.

Precision is calculated for the final retrieval set using the standard formula of

number of relevant passages retrieved (absolute) / number of passages retrieved

Recall is calculated for the final retrieval set using the standard formula of

number of relevant passages retrieved (absolute) / total number of relevant passages available

The *ranking balance point* (R) for each retrieval set (not just the final one) is calculated by

$$\frac{\sum_{i=1}^{n} i * relevance_i}{\sum_{i=1}^{n} relevance_i}$$

where n = number of passages in the retrieval set
i = position of the passage in the retrieval set
$relevance_i$ = relevance weight of passage i

This calculates where the midpoint of the relevant passages lies, accounting for the relevance weight. The earlier in the retrieval set the relevant passages occur, the smaller their midpoint. For example, consider a retrieval set of five passages of which the first two are very relevant (weight = 2), the next two irrelevant (weight = 0), and the last passage somewhat relevant (weight = 1). The ranking balance point for this set would be:

$(1*2) + (2*2) + (3*0) + (4*0) + (5*1) / 6 = 1.83$

The *random balance point* (R) for each retrieval set is calculated by (n+1)/2 where n is the number of passages in the retrieval set. A random distribution of relevant passages in the set would have the midpoint (M) of the retrieval set as the balance point. Therefore, the random balance point for the set of five passages in the previous example would be 3.

The *best case balance point* (BC) for each retrieval set is calculated by applying the ranking balance point formula to the case where all very relevant passages preceded all somewhat relevant passages which in turn preceded all non-relevant passages in the set. In this case, the ranking balance point would be:

$(1*2) + (2*2) + (3*1) + (4*0) + (5*0) / 6 = 1.5$

The *normalized ranking balance points* were calculated from the ranking balance points by moving the random balance point to 0 and adjusting the range so that the best case balance point fell on 1, and the worst case balance point at -1. The normalization performed was:

Normalized ranking balance point (NR) = $(M - R) / (M - BC)$.

For the example retrieval set, the normalized ranking balance point would be:

$(3 - 1.83) / (3 - 4.5) = 0.78$.

Summaries Calculated for Each System

For each system the means calculated were:

- number of queries per question
- time per question (seconds)
- number of relevant passages (user) per question
- user precision
- number of relevant passages (from absolute retrieval set)

- precision
- recall

For each ranking algorithm (the expert system's, and randomness) the normalized balance points were calculated.

3.3 Results

The means were compared to determine if their differences were statistically significant. Pairwise two-tailed t-tests were performed. A difference was considered significant if its probability of occurring due to chance was less than 5% at the 95% confidence level (a 10% chance at the 95% confidence level was considered marginally significant). Pairs of means with statistically significant differences are flagged with asterisks.

Search Effectiveness

All three systems retrieved comparable numbers of relevant passages. Whereas there seemed to be higher recall with the thesaurus, shown by a mean of 7.688 compared to a mean of 7.292 with the expert system, this difference was not significant ($p = 0.5333$).

- number of relevant passages (user)
  per question
  - user alone 7.375
  - user and thesaurus 7.688
  - user and expert system 7.292

All three systems produced comparable precision, based on the subject's relevance judgements.

- user precision
  - user alone 0.763
  - user and thesaurus 0.786
  - user and expert system 0.761

All three systems retrieved approximately the same number of passages from the absolute retrieval set.

- number of passages from absolute retrieval set
    - user alone 5.521
    - user and thesaurus 5.708
    - user and expert system 5.729

Recall was comparable across all three systems. There was a slight improvement in recall for the user and expert system configuration, but the advantage over the user-alone configuration was not significant ($p < 0.6988$).

- recall
    - user alone 0.364
    - user and thesaurus 0.368
    - user and expert system 0.379

The user and expert system configuration produced marginally significant improvements in precision when compared with the user-alone configuration.

- precision
    - user alone 0.530 * ($p < 0.0817$)
    - user and thesaurus 0.576
    - user and expert system 0.604 *

Search Efficiency

The expert system was not significantly slower than the other two systems. However, the user was marginally significantly slower when using a thesaurus. However, MICROARRAS was being executed by a Sun 4 with the user-expert system configuration resulting in approximately a doubling of its speed.

- mean time per question (seconds)
  - user alone 474.5 * ($p < 0.101$)
  - user and thesaurus 571.5 *
  - user and expert system 539.8

The expert system improved search efficiency, as measured by number of user queries over both the user alone and user plus thesaurus.

- number of queries per question
  - user alone 4.833 * ($p < 0.0001$)
  - user and thesaurus 5.458 ** ($p < 0.0001$)
  - user and expert system 2.354 *,**

Ranking

The expert system ranked relevant documents more highly than would be predicted by randomness. The expert system's ranking was compared to a random distribution for 74 sets of retrieved passages.

- balance points
  - random 5.00 * ($p < 0.0165$)
  - expert system 4.53 *
- normalized balance points (on range of -1 to +1)
  - random 0.000 * ($p < 0.0025$)
  - expert system 0.195 *

3.4 Analysis

The first hypothesis, that the expert system can improve the search effectiveness for a novice user was partially supported by this study. The expert system produced marginally significant improvements in precision, and seemed to indicate improvements in recall, but these results were not significant. Providing the online thesaurus produced no improvement in search effectiveness.

The improvements in precision may result from the expert system applying better broadening techniques. The subjects, when searching alone, would often stop with a very broad query and examine a large set of retrieved passages (over fifteen) looking for relevant information. This type of strategy results in the lower precision observed when the subjects search on their own.

However, this browsing strategy also accounts for the ability of the subjects to produce recall comparable to the expert system. For example, in two questions with large absolute retrieval sets the subjects were able to retrieve, on average, 10 and 10.25 relevant passages on their own compared with the expert system's retrieval of 8 and 7.75 passages respectively. By using a target number of 10 for these broader questions, the expert system was operating at a disadvantage. More relevant information was easily found, judging by the high recall of the subjects, but the expert system did not even attempt to further broaden the query.

The second hypothesis, that the expert system can improve the search efficiency of novice searchers, was supported. Using the expert system significantly reduced the number of queries subjects needed to answer a given question. Subjects required fewer than half as many queries per question on average versus systems in which the user queried without it, a substantial improvement. The expert system reduced the amount of user effort required by decreasing the number of queries a user needs to design to express their information needs. If efficiency is measured in terms of total user time the expert system fares less well. The expert system was not significantly slower than either of the other two systems but it was necessary to run MICROARRAS on a faster machine to achieve this. However, this version of the expert system was designed with correctness rather than efficiency in mind, and there are several ways that it could be sped up.

Allowing the subjects to access the online thesaurus actually decreased the subjects' efficiency. They took significantly more time than when they searched on their own, and required no fewer queries. This allows us to conclude that the improvement in efficiency seen above was due to the expert system's searching knowledge base, not just the provision of an online thesaurus.

The third hypothesis that the expert system could rank passages in decreasing order of relevance was supported. Although the expert system did present relevant passages significantly earlier than would be predicted by randomness, the improvement was not large enough to be considered truly successful. The current algorithm needs to be evaluated with different weights or a somewhat different algorithm needs to be tried in order to further improve the ranking function. Decreasing the query term weights more quickly as the query terms move farther from the original may improve the ranking by placing more emphasis on the user's original search terms. Using a more sophisticated closeness factor, one that took into account to how many words apart the search terms were in the passage, as well as sentence and paragraph measures considered in this version, could also lead to improved ranking.

3.5 Questionnaire

The twelve subjects were asked which features of the expert system they liked best. The automatic addition of terms from the thesaurus was the most frequently mentioned (8 subjects), whereas the automatic context adjustment was the second most popular feature (3 subjects). Many subjects (8) mentioned the decreased amount of work needed to perform a search, with three of them specifically mentioning that they did not have to think as much. Other features mentioned which decreased the user effort were the simplified syntax, decreased typing, and the fewer queries to remember.

System slowness was the feature most disliked (6 subjects). Although the amount of time necessary to answer a question was no greater with the expert system (see Section 4.3.1),there was less work for the user so time seemed longer. The other main complaints concerned the user interface. The subjects were fairly evenly split between wanting the system to proceed more automatically, with less prompting from them (4 subjects), whereas others wanted the system to explain what it was doing and/or allow the user to direct it (5 subjects). These comments lead to the conclusion that if a usable system is to be built based on the success of this research prototype, the execution of the system must be sped up and more work on interface design is needed.

Almost all the subjects (10) found the user-expert system version the easiest to use, with the remaining two subjects split between the other two versions. Not surprisingly, given the comparable effectiveness of the three systems, the subjects were split on which system they felt gave the best results. Three voted for the user-alone version, two for the user-thesaurus, and three for the expert system. Three said it was a tie between the user-thesaurus and the expert system, and one abstained.

4 CONCLUSIONS

I have designed, implemented, and evaluated an expert system that automatically reformulates contextual Boolean queries for full-text information retrieval. Whereas more research is necessary to develop a better search assistant, I have demonstrated that a domain-independent online search assistant can be developed now. This is important because if more people can successfully search online textbases, and they can do so with less effort, the information stored in these textbases will become more widely disseminated.

Running the experiment suggested several possible refinements to the system. The experimental subjects had many useful comments, the bulk of which dealt with the desire for a more sophisticated user interface. Desirable changes include: provision of a non-Boolean query language; allowing users to adjust the amount of system interaction; having the user specify the type of search desired, rather than having him give a specific target number; and increasing the speed of the system by improving the way the expert system uses MICROARRAS. Improvements to the searching knowledge base were suggested by observing the expert system in use. Specifically, the order in which the search tactics are applied needs further investigation. Additionally, more narrowing techniques are needed, and expansion of multi-word phrases could be handled better. The thesaurus used for this textbase was developed manually. Research is needed in how to automatically this process. Finally, other ranking algorithms should be investigated.

REFERENCES

Bates, Marcia J., "Information Search Tactics", *Journal of the ASIS*, Vol. 30, No. 4, 1979. pp. 205-214.

Blair, David C., and Maron, M.E., "An Evaluation of Retrieval Effectiveness for a Full-Text Document-Retrieval System", *Communications of the ACM*, Vol. 28, No. 3, March 1985. pp. 289-299.

Blaauw G.A. and Brooks, Frederick P. Jr., *Computer Architecture, Volume 1-Design Decisions*, Draft, Spring 1987.

Borgman, Christine L., "Why Are Online Catalogs Hard to Use?", *Journal of the ASIS*, Vol. 37, No. 6, November 1986. pp. 387-400.

Borgman, Christine L., "Individual Differences in the Use of Information Retrieval Systems: Some Issues and Some Data", *Proceedings of the Tenth Annual International ACMSIGIR Conference on Research & Development in Information Retrieval*, C.J. von Rijsbergen and C.T. Yu (ed.), ACM Press, 1987. pp. 61-69.

Brajnik, G., Guida, G., and Tasso, C., "IR-NLI II: Applying Man-Machine Interaction and Artificial Intelligence Concepts to Information Retrieval", *Proceedings of the Eleventh Annual International ACMSIGIR Conference on Research & Development in Information Retrieval*, Y. Chiaramella (ed.), ACM Press, 1988. pp. 387-399.

Chiaramella, Y. and Defude, B., "A Prototype of and Intelligent System for Information Retrieval: IOTA", *Information Processing & Management*, Vol. 23, No. 4, 1987. pp. 285-303.

Croft, W. B. and Thompson, R. H., "I3R: A New Approach to the Design of Document Retrieval Systems",*Journal of the ASIS*, Vol. 38, No. 6, November 1987. pp. 389-404.

Fenichel, C. H., "Online Searching: Measures that Discriminate Among Users with Different Types of Experience", *Journal of the ASIS*, Vol. 32, No. 1, January 1981, pp. 23-32.

Fox, Edward A., Weaver, Marybeth T., Chen, Qi-Fan, and France, Robert K.,"Implementing a Distributed Expert-Based Information Retrieval System", *Proceedings of RIAO 88*, M.I.T., March 1988. pp. 708-726.

Gauch, Susan, "An Expert System for Searching in Full-Text", Doctoral Dissertation, Dept. of Computer Science, *Tech. Report* #TR89-050, University of North Carolina, Chapel Hill.

Gauch, Susan, and Smith, John B., "An Expert System for Searching in Full-Text", *Information Processing &Management*, Vol. 25, No. 3, 1989. pp. 253-263.

Gauch, Susan, and Smith, John B.,"Query Reformulation Strategies for an Intelligent Search Intermediary", Proceedings of the Annual AI Systems in Government Conference, Washington, D.C., March 1989.

Oldroyd, B. K., "Study of strategies used in online searching 5: differences between the experienced and the inexperienced searcher", *Online Review*, Vol. 8, No. 3, 1984. pp. 233-244.

Pollitt, A.S., "A `Front-End' System: An Expert System as an Online Search Intermediary", *ASLIB Proceedings*, Vol. 36, No. 5, May 1984. pp. 229-234.

Shoval, Peretz, "Principles, Procedures and Rules in an Expert System for Information Retrieval", *Information Processing &Management*, Vol. 21, No. 6, 1985. pp. 475-487.

Smith, John B., Weiss, Stephen F., and Ferguson, Gordon J., "MICROARRAS: An Advanced Full-Text Retrieval and Analysis System", *Proceedings of the Tenth Annual International ACMSIGIR Conference on Research & Development in Information Retrieval*, C.J. von Rijsbergen and C.T. Yu (ed.), ACM Press, 1987. pp. 187-195.

Smith, Phillip J., Shute, Steven J., and Galdes, Deb, *Proceedings of the Twelfth Annual International ACMSIGIR Conference on Research & Development in Information Retrieval*, N. J. Belkin and C. J. van Rijsbergen (ed.), ACM Press, 1989. pp. 3-10.

Teskey, Niall, "Extensions to the Advanced Interface Management Project", *OCLC Research Review*, July 1987. pp. 1-3.

Tong, Richard M., Applebaum, Lee A., Askmann, Victor N., and
Cunningham, James F., "Conceptual Information Retrieval using RUBRIC",*Proceedings of the Tenth Annual International ACMSIGIR Conference on Research & Development in Information Retrieval*, C.J. von Rijsbergen and C.T. Yu (ed.),ACM Press, 1987. pp. 247-253.

Vernimb, Carlo, "Automatic Query Adjustment in Document Retrieval", *Information Processing & Management*, Vol. 13, No. 6, 1977. pp. 339-353.

Vickery, A., and Brooks, Helen M., "PLEXUS-The Expert System for Referral", *Information Processing & Management*, Vol. 23, No. 2, 1987. pp. 99-117.

Williams, P.W., "A Model for an Expert System for Automated Information Retrieval", *Proceedings of the 8th International Online Meeting*, 1984. pp. 139-149.