

# The Use of Dynamic Contexts to Improve Casual Internet Searching

GONDY LEROY, ANN M. LALLY, and HSINCHUN CHEN  
The University of Arizona

---

Research has shown that most users' online information searches are suboptimal. Query optimization based on a relevance feedback or genetic algorithm using dynamic query contexts can help casual users search the Internet. These algorithms can draw on implicit user feedback based on the surrounding links and text in a search engine result set to expand user queries with a variable number of keywords in two manners. Positive expansion adds terms to a user's keywords with a Boolean "and," negative expansion adds terms to the user's keywords with a Boolean "not." Each algorithm was examined for three user groups, high, middle, and low achievers, who were classified according to their overall performance. The interactions of users with different levels of expertise with different expansion types or algorithms were evaluated. The genetic algorithm with negative expansion tripled recall and doubled precision for low achievers, but high achievers displayed an opposed trend and seemed to be hindered in this condition. The effect of other conditions was less substantial.

Categories and Subject Descriptors: H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—*Information filtering, query formulation, relevance feedback, retrieval models, search process*; H.3.5 [Information Storage and Retrieval]: Online Information Services—*Web-based services*

General Terms: Algorithms, Human Factors, Performance

Additional Key Words and Phrases: Information retrieval, personalization, Internet, genetic algorithm, relevance feedback, automatic query expansion, implicit user feedback

---

## 1. INTRODUCTION

Although a growing amount of information is made available on the Internet, search engines do not provide sufficiently sophisticated methods to help casual Internet users access relevant pages. Casual users are defined as the group of users who regularly search for information on different topics but who are not search experts. Many of these users' domain knowledge, experience, and

---

This research was sponsored by the following grant: NSE Digital Library Initiative-2, "High-Performance Digital Library Systems: from Information Retrieval to Knowledge Management," IIS-9817473, April 1999-March 2002.

Authors' addresses: G. Leroy: Claremont Graduate University; email: gondy.leroy@cgu.edu; A. Lally and H. Chen: Department of Management Information Systems, The University of Arizona, McClelland Hall, Room 430, 1130 E. Helen St., Tucson, AZ 85721; email: alally@u.washington.edu; hchen@eller.arizona.edu.

Permission to make digital/hard copy of part or all of this work for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication, and its date appear, and notice is given that copying is by permission of ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or a fee.

© 2003 ACM 1046-8188/03/0700-0229 \$5.00

success in searching the Internet are varied [Meyer et al. 1997; Pitkow and Kehoe 1996] and their search results frequently are poor [Nordlie 1999; Thury 1998].

One reason for low success in Internet searching is using a small number of keywords: around two [De Lima and Pedersen 1999; Ross and Wolfram 2000; Spink et al. 2001; Toms et al. 2001], too few to retrieve a subset of relevant pages from a collection of millions. Fortunately, most users (77%) engage in multiple search sessions on the same topic, using the same search engine [Sullivan 2000], a behavior also observed in database searches [Spink 1996]. We believe we can extract valuable information regarding the implicit user interests from their behavior during consecutive searches. Based on this information, we can expand user queries or limit results.

The goal of this study was to discover whether relevance feedback or genetic algorithms can improve Internet searching by expanding user queries based solely on the context provided by the search results users review. Research on automatic query expansion reveals that both relevance feedback and genetic algorithms based on user feedback are good candidates for improving search results. In many applications where texts are peer-reviewed documents, both algorithms have produced substantial increases in performance.

Although the Internet is much noisier than databases that contain peer-reviewed documents and the information available per user search session is sparse, the magnitude of the problem warrants investigating the usefulness of different algorithms. This is a novel approach because we help casual users with different levels of expertise and a one-time need for specific information. Ideally, the retrieval procedure should not depend on any additional effort by the user. Our approach does not require users to submit a profile or to exert any additional effort, and does not invade privacy by gathering and retaining user information. Furthermore, by using only information of a particular search to build a dynamic query context, user interest can change for every search without affecting the algorithms. Additionally, we believe that not all users need help since they differ substantially in search expertise. Therefore, it is important to discover how users with different performance levels can be helped further. We chose relevance feedback and the genetic algorithm because both have been used very efficiently to improve information retrieval.

In the following, we review related work on how query expansion improves information retrieval. We describe relevance feedback and genetic algorithms and explain why both are viable candidates for improving online searching with search engines. Subsequently, we describe our specific adaptation of the algorithms and how a query context is dynamically established, followed by our research questions, user study, discussion, and conclusions.

## 2. BACKGROUND

### 2.1 Query Expansion

Many Internet queries consist of only a few keywords and the results obtained with them are not always satisfactory. These results can be improved by

expanding the query with additional search terms. Results from the tenth Text REtrieval Conference (TREC 2001), for example, indicate that the top ranked runs used some form of query expansion based on some type of relevance feedback [Hawking and Craswell 2001]. Queries can be expanded in different manners. With manual query expansion, users indicate which terms should be used for expansion. With automatic query expansion, a system selects the terms for expansion.

Query expansion depends on user feedback. Explicit user relevance feedback is based on users indicating which results of a search are relevant. Based on this evaluation, terms from the relevant documents can be used for query expansion either automatically or based on users' choice. Implicit or pseudo feedback deduces the evaluation from the user behavior without actually asking the user for the feedback. The terms are often automatically added to a user query or used to modify a user query.

Early research provides indications of the optimal number of required expansion terms. Salton and Buckley [1990], for example, used a small biomedical collection with 1033 documents and a larger computer engineering collection with 12,684 documents. They discovered that full expansion was modestly better than expansion based on the most common terms. However, when processing speed is an issue they advised the use of only the most common terms. Full expansion with all relevant terms and those from the first nonrelevant document formed the best combination. Harman [1988; 1992] used the Cranfield collection with 1400 documents to study vector-processing methods and found performance improvements of over 100%. She also found that adding a selected set of relevant terms (20) was better than adding all relevant terms. The terms were selected from a list containing words found in relevant documents and based on statistical techniques involving term and document frequency. Later, Magennis and van Rijsbergen [1997] found that for automatic query expansion, the optimal cutoff point could be as low as six additional terms instead of 20. They used the term *ranking function* from Robertson and Sparck Jones [1976] for their term selection.

Query expansion has been studied extensively at the Text REtrieval Conferences (available online at <http://trec.nist.gov/>) for several years. In general, it was found that relevance feedback increases performance of information retrieval systems. Amati et al. [2001] reported on automatic query expansion based on probabilistic distributions of the terms in the document collection and found expansion to be beneficial, especially for the precision of results. However, they found that the parameters need to be tuned for different collections. In the Interactive Track, Bodner and Chignell [1998] found that explicit relevance feedback was critical in boosting their system's performance from below to above average performance. In their case, the users indicated when they encountered relevant documents during their search. Yang and Maglaughlin [Yang and Maglaughlin 1999; Yang et al. 1998] tested the difference of relevance feedback based on a complete document or a passage in the document in TREC-7 and TREC-8. The passage-based feedback system performed better than the document-based feedback system in TREC-7 but not in TREC-8. Koenemann and Belkin [1996] compared the performance of novice users with

three types of relevance feedback: fully automatic, automatic but showing the terms, and automatic with the possibility for users to intervene. Performance was 17 to 34% higher with relevance feedback, especially when users could modify the expanded query. The authors also reported substantial individual differences between their users.

In a real setting, users seldom request query expansion: 5% of the time or less according to a study by Jansen et al. [2000]. It is therefore our purpose to use implicit feedback for automatic query expansion so as not to burden a user with additional tasks. Several researchers have looked at potential implicit measures of user interest, such as the number of hits a Web page gets, time spent on a page, or printing or bookmarking a page [Claypool et al. 2001; Fuller and de Graaff 1996; Lai and Yang 2000]. The results obtained with automatic query expansion based on implicit feedback have been mixed. These implicit measures of relevance are often inaccurate or unavailable during casual Web searches. Hit count may be inaccurate because Web browsers cache Web pages, which results in fewer hits. Time spent on a Web page may be an inaccurate measure when many graphics have to be downloaded, or with changing network speeds. Printing and bookmarking information is valuable over a prolonged period of observation, but the information is often unavailable from casual searches.

Automatic query expansion is often developed with a technique called *local feedback* [Attar and Fraenkel 1977] or *local document analysis* [Xu and Croft 1996], which provides text sources from which expansion terms can be automatically extracted. Attar and Fraenkel [1977] considered all top ranked results to be relevant. However, Xu and Croft [1996] found that the proportion of actually relevant documents in the top ranked documents affected the results, with a higher proportion resulting in better performance. Later, they used local context analysis [Xu and Croft 2000] and based query expansion on the cooccurrence of concepts with query terms within documents. Finkelstein et al. [2002] successfully based query expansion on the words surrounding query terms in documents users read before searching for additional information. Vogt's [2000] results from TREC-9 showed that when it is possible to measure the reading time, this time normalized by the document length might indicate relevant documents since the distribution of reading times for relevant and nonrelevant documents differs. Budzik and Hammond [2000] used the documents users are working on, for example, papers they are writing, to provide contextual information to find additional information for an anticipated information need. Based on the contextual information, they retrieved several new relevant Web pages for their users. White et al. [2002] used the time spent on document summaries as an indication of interest. They reranked Web pages based on a combination of user search terms and terms taken from relevant documents. Users did not have an increased perception of task completion with implicit feedback, but it appeared to help inexperienced users search.

Belkin et al. [1999] compared automatic and manual query expansion for the TREC-8 interactive task and found no differences in performance or preference by the users. They concluded that automatic query expansion was the preferred method since it required less user effort. Later, White et al. [2001] argued

in TREC-10 that implicit feedback can substitute for explicit feedback. They used the request for a document summary as an indication of implicit user interest and found that there were no differences in time spent or perceived task completion between systems based on explicit or implicit feedback. We devised a related implicit measure that we believe can provide a good indication of the user interests: examining the links followed or ignored by the users. If a user follows a link, something of interest must appear in the title or description of that page. If the user ignores a link, nothing interesting is presented. When a link is followed, we consider the text surrounding the link as relevant but not the underlying document since the user has not yet read this. This information is easily captured by storing text in separate categories determined by a user's decision to follow or ignore a link. It does not intrude on user privacy, nor does it require any additional user effort.

## 2.2 Relevance Feedback Algorithm

The relevance feedback algorithm was introduced in the mid-1960s as a technique for controlled and automatic query reformulation. The main idea behind relevance feedback is to represent user interest or lack of interest and use this information to enhance user queries. The original feedback technique was designed to be used with queries and documents mapped to multidimensional vectors. Vector processing methods calculate the similarity between a document ( $D$ ) and a query ( $Q$ ) as the inner product between the two vectors. Many variations exist, but the basic similarity formula, where  $q_i$  and  $d_i$  represent the weights of terms in the query and documents, is the following:

$$Sim(D, Q) = \sum_{i=1}^t d_i \cdot q_i.$$

The purpose of relevance feedback is to use relevant and nonrelevant document sets to modify a user query, making it more similar to the set of relevant documents. The new reformulated query is expected to retrieve more documents similar to those initially identified as being relevant.

There are three main variations to modify the query vector. The first variation is known as Ide-regular [Ide 1971; Ide and Salton 1971]: terms found in previously considered relevant documents are added to the original query vector, terms found in nonrelevant documents are subtracted from the original query vector, and negative weights are replaced with a zero. The second variation is known as Ide dec-hi [Ide 1971; Ide and Salton 1971]. Here, terms from the relevant documents are added to the original query, and terms from the first nonrelevant document are subtracted from the original query vector. The third variation is that used by Rocchio [1971]: terms from all relevant documents are added and terms from all nonrelevant documents are subtracted, but additional terms have a lower weight than original terms.

Besides the vector representation methods described above, the probabilistic method is another popular feedback method. It was developed somewhat later and is based on the distribution of terms in a document collection. We refer to work by Robertson and Sparck Jones [1976] and van Rijsbergen [1979]

for details. With large Web collections, both probabilistic and vector space approaches perform equally well with the TREC ad hoc Web tasks [Hawking and Craswell 2001]. Others have found that the vector processing methods outperformed the probabilistic methods [Salton and Buckley 1990].

It is our purpose to use implicit feedback so as not to burden a user with additional tasks. We concentrate on an approach closely related to the vector processing methods and modify queries with terms from relevant and nonrelevant contexts. The contexts are represented by the text users see when deciding to follow or ignore a link, that is, the title and text fragment of a document displayed by the search engine.

### 2.3 Genetic Algorithm

Genetic algorithms were developed in the early 1990s and belong to the class of evolutionary programs [Michalewicz 1992]. They are stochastic algorithms modeled after natural evolution based on survival of the fittest and genetic inheritance [Michalewicz 1992]. The problem or function to be evaluated has to be mapped to genetic variables such as individuals or chromosomes, genes or bits, a population, and a fitness function. A population improves over several generations based on the reproduction and recombination of the fittest individuals.

Genetic algorithms have been used in information retrieval in different manners. They have been used to build user profiles by monitoring people's browse behavior over time [Chen et al. 2001; Nick and Themis 2001]. When used to directly modify a user query, the document and query are represented as vectors comparable to the relevance feedback methods described above. Each individual represents a query vector and the individual genes can represent the weights of the keywords or the presence or absence of keywords. Yang and Korfhage [1993] successfully used the genetic algorithm with different collections to weight keywords. Chen et al. [1998b] used a database with 8000 records and found that the genetic algorithm outperformed relevance feedback, ID3, and simulated annealing. Chen et al. [1998a] later compared a genetic algorithm with best first search for spidering relevant Web pages and found that recall was higher with the genetic algorithm; precision was equal to best first search. An additional interesting finding was that the Web pages from the two algorithms were not overlapping but were largely complementary.

Genetic programming is an extension of genetic algorithms used in information retrieval. Although the underlying algorithm is basically the same, the internal data structures that comprise the individuals are more complex, for example, trees instead of numbers. In Kraft et al.'s [1994] implementation, the individuals represent Boolean queries. Each gene represents a subset of a query, such as (*and*  $t_1, t_2, t_3$ ), which represents the conjunction of the three terms  $t_1$ ,  $t_2$ , and  $t_3$ . These researchers experimented with a small document collection of 483 abstracts taken from the *Communications of the ACM* and tested different seeding strategies and fitness functions. Although unable to draw conclusions about different fitness functions, they learned that seeding the initial population with terms taken from a predefined set of relevant documents gave better

results than seeding with terms based on the distribution of the terms in the complete collection. Unfortunately the authors did not provide a user baseline, and it is impossible to compare the algorithm queries with actual user queries.

In other information retrieval implementations, individuals do not represent queries but matching functions. Fan et al. [2000] used partial weighting functions as genes such that an individual represented a matching function. The authors tested their system on the TREC-4 database collection, which contains 55,554 documents. Since this collection includes relevance evaluations for queries, they trained their algorithm on a single training set with 2200 documents that included relevant documents for all 50 queries. They then tested the 20 best individuals or matching functions for each query on the complete dataset (including the test set) and found that performance increased dramatically for all queries. This approach is very useful if all subsequent user queries can be automatically defined as being similar to the ones used for training. Completely new or different queries will need a new training round. Pathak et al. [2000] also used genetic programming to find an optimal matching function. They let the genetic algorithm combine and weight four existing matching functions, that is, Cosine, Jaccard, Dice, and Overlap, into a single function, and used van Rijsbergen's [1979] performance measure ( $E$ ), a combination of both recall and precision, as the fitness function. Using the Cranfield collection as their test bed, they found that the combined fitness functions performed better than the individual matching functions.

### 3. SYSTEM IMPLEMENTATION

As stated above, our goal was to test both relevance feedback and a genetic algorithm for their potential to modify user queries, thereby improving the results returned by a search engine and helping users accomplish a task. Since batch evaluations are not necessarily good predictors of performance with real users [Hersh et al. 2000a, 2000b], we built our algorithm with individual user interactions in mind and we also performed the evaluation in a real user setting. The advantage is that we could see the interaction of actual users with the algorithms when solving real tasks. The disadvantage is that less data can be collected since it is impractical to test hundreds of users. The following gives an overview of what constitutes a user session, how implicit user feedback is collected to establish and update two types of contexts dynamically, and how each algorithm modifies user queries based on this data.

#### 3.1 Data Collection During User Searches

When a user needs to find information regarding a particular topic he or she starts searching by typing keywords and selecting "search" on our Java interface, as in any search on the Internet. A connection to a search engine is established and the results of the first search are displayed to the user. These results are the first 10 Web pages with their titles and descriptions returned by the search engine. Our algorithms never modify the first user query, since no user feedback is as yet available and since we do not predefine a set of documents as being relevant. Instead, the relevant and nonrelevant contexts are

built on the fly for each search session for each subject. These contexts are based on the titles and descriptions users see, not the underlying documents. The system attempts to expand each second and subsequent query. For this system implementation, we do not provide users the option to start over with an “unmodified” query since we envision only a short search session per query. Later development can provide this option.

A search session is a set of consecutive searches by a user to find information on the same topic. When a user follows a link to a Web page, the title and description are categorized as relevant, and links not followed are categorized as nonrelevant. The rationale behind this is that if a user follows a link something in the title or description is of interest to him or her. The category having followed links contains the implicit positive feedback and represents the relevant context. The user keywords are also added to this category. The category having ignored links contains the implicit negative feedback or the nonrelevant context. The words and their occurrence frequency are retained for both contexts separately. Since users engage in multiple searches, the contexts change with every search, so both are continuously updated during a search session. The algorithms will use words from these sets to modify subsequent user queries. The additional keywords can be added in either of the following ways:

$$Q_{new} = Q_{user} + \sum_{\substack{\text{top-system} \\ \text{relevant}}} D_i,$$

$$Q_{new} = Q_{user} - \sum_{\substack{\text{top-system} \\ \text{non-relevant}}} D.$$

Since Xu and Croft [1996] found that the proportion of relevant documents in the top-ranked documents affects the results, with a higher proportion resulting in better performance, we use implicit user feedback to differentiate between relevant and nonrelevant documents in the top-ranked results. We assume that this will increase the proportion of relevant and nonrelevant titles and descriptions correctly assigned to the relevant and nonrelevant contexts. The relevance of the system-selected terms is based on the frequency of the terms in the respective relevant and nonrelevant sets and additionally, for the genetic algorithm, the resulting similarity to the relevant and nonrelevant contexts. Words appearing in both relevant and nonrelevant contexts are removed from each and stored in an additional word set for use by the genetic algorithm. Both contexts serve as key term pools for query expansion. They also serve as the comparison base for the genetic algorithm’s fitness function. We will use the most frequently occurring concepts of the relevant and nonrelevant contexts for query expansion.

### 3.2 Relevance Feedback Implementation

Our relevance feedback algorithm is adapted for the Internet environment. Since we do not have an exhaustive list of all possible keywords that would

allow us to use the traditional vector approach, we add the actual keywords to the user query. We tested different expansion numbers during preliminary testing. Initially we added 10 keywords to the user's queries, but, too often, no results were found for these queries. Adding only one or two keywords seldom changed the query in a substantial way. We decided to add five keywords to the user query because this modifies a query while still retrieving documents. The most common terms, that is, terms from either the relevant or nonrelevant context with the highest frequency count, are selected. If terms have identical frequencies, the first to have been encountered is chosen. Terms taken from the relevant context are added to the user keywords with a Boolean *and*; terms taken from the nonrelevant context are added with a Boolean *not*.

The terms are automatically added to the user queries and users do not see the query submitted to the search engine. However, during preliminary testing we noticed that users became very suspicious when only a limited number of results were returned, even if these results were very good. To avoid users being biased based solely on the number of results returned, we returned to using queries with user-only keywords whenever there are fewer than five Web page links in the returned results. Users were not told when this happened. The results were shown in the same manner as the search engine returned them for both modified and unmodified queries and we did not eliminate previously seen pages from the results.

### 3.3 Genetic Algorithm Implementation

Similarly, our genetic algorithm adds five keywords to the user submitted keywords for the same reason and to maintain comparability with our relevance feedback implementation. Keywords are selected in the following steps:

- Initial population*: Each population consists of 10 individuals. An individual represents a query and has a certain number of open slots, genes, to be filled with keywords. The maximum number of genes is the number of user keywords plus five additional terms. In the initial population, each individual contains all user keywords. The algorithm adds five search terms. These terms do not need to be unique. The selection of terms for the first population is based on their frequency in the relevant or nonrelevant context; terms with a higher frequency have a better chance of being selected. As with relevance feedback, the system-suggested terms are selected from either the relevant or the nonrelevant context.
- Fitness function*: Each individual is sent to the search engine formatted as a query and the first 10 results returned by the search engine are retrieved. All words, except stop words, are extracted from this result set. All words from text associated with followed or relevant links up to and including search  $x$  are part of the set  $S_{Rx}$ . Words from text associated with ignored links up to and including search  $x$  are part of the set  $S_{Nx}$ . There is an additional set of words containing stop words:  $S_S$ .

There are three contexts based on the word sets that are used by the genetic algorithm. Each context is continuously updated with each search the user

performs and used to modify and evaluate search  $x + 1$ . The relevant context ( $C_{Rx}$ ) contains all words ( $w_i$ ) from the relevant word set that are not part of the nonrelevant word set or the stop word set. The nonrelevant context ( $C_{Nx}$ ) is similarly built for nonrelevant words ( $w_i$ ). The additional context ( $C_{Ax}$ ) contains all words ( $W_k$ ) that appear in both the relevant and nonrelevant word sets but not in the stop word list.

$$\begin{aligned} C_{Rx} &= \forall w_i \mid w_i \in S_{Rx} \wedge w_i \notin S_{Nx} \wedge w_i \notin S_S, \\ C_{Nx} &= \forall w_j \mid w_j \notin S_{Rx} \wedge w_j \in S_{Nx} \wedge w_j \notin S_{Sx}, \\ C_{Ax} &= \forall w_k \mid w_k \in S_{Rx} \wedge w_k \in S_{Nx} \wedge w_k \notin S_{Sx}. \end{aligned}$$

An individual or query's fitness score is a Jaccard score based on the words in the results and their similarity to the relevant and nonrelevant context.  $R_{ix}$  represents the words found in the search engine results for individual  $i$  during search  $x$ .  $J_{Rix}$  represents the similarity of the individual to the relevant context;  $J_{Nix}$  represent the similarity of the individual to the nonrelevant context.

$$\begin{aligned} J_{Rix} &= J_R(R_{ix}, C_{Ri}) = \frac{\#(R_{ix} \cap C_R)}{\#(R_{ix} \cup C_R)}, \\ J_{Nix} &= J_N(R_{ix}, C_{Ni}) = \frac{\#(R_{ix} \cap C_N)}{\#(R_{ix} \cup C_N)}. \end{aligned}$$

An individuals' final score ( $J_{ix}$ ) combines both Jaccard scores by subtracting the nonrelevant score from the relevant score and normalizing the result:

$$J_{ix} = \frac{(J_{Rix} - J_{Nix}) + 1}{2}.$$

The population fitness ( $J_{pop}$ ) is the sum of all individual fitness scores:

$$J_{pop} = \sum_i J_{ix}.$$

- Reproduction*: We use the imaginary roulette wheel to select individuals for the next generation. The probability of an individual being selected for reproduction is the proportion of its fitness to the population fitness.
- Recombination*: Single-point crossover and mutation are used for recombination. One crossover point is randomly selected to swap the genes of two individuals following this point. Keywords for mutation are randomly selected from the additional context ( $C_{ax}$ ) (described above, i.e., containing the words that belonged to both followed and ignored links). The next generation of individuals is simultaneously sent off to the search engine.
- Convergence*: Because individuals represent complete queries, we use a centroid method for convergence. It is sufficient reason to continue the cycle if one individual with its associated results improves from one generation to the next. These steps are repeated until there is no more improvement of the best individual from one generation to the next.

As with relevance feedback, the user keywords are resubmitted unmodified when the algorithm does not provide five or more Web pages in the result set and results are shown as they are returned by the search engine.

#### 4. RESEARCH QUESTIONS

Two main interests led to this research. The first was related to the usefulness of relevance feedback and genetic algorithms to expand Internet user queries. In many studies the algorithms were used to access many high-quality documents. However, our approach was different because there was less information available with actual users and the quality of the Web-based information was lower. There was less information, given that only the title and description of the links followed by the user were available. Additionally, the quality of the available Web text was generally lower than the quality of peer-reviewed documents. Even with these two disadvantages, many improvements reported are so substantial that we believe the algorithms to be strong enough to improve user queries and, therefore, search engine results. Since users submit only a limited number of keywords and the Internet contains millions of accessible Web pages, adding additional keywords to a query will narrow the set of suitable documents. We expected both algorithms to have a beneficial effect on performance (as measured by precision and recall) because both expand a query with additional search terms. We also believed that explicitly adding filtering terms that are unwanted in the results (negative query expansion) to the user-supplied terms would increase precision more than would adding extra required terms (positive query expansion). With negative query expansion, a search is based on user keywords and the results are filtered by additional terms judged nonrelevant. We believed this would focus the results more than adding relevant but probably related terms. Although the genetic algorithm can be seen as more powerful than our implementation of relevance feedback, we believed that each algorithm has its strengths. We believed that the genetic algorithm would excel with positive expansion because it can find a few good query terms that characterize the search query optimally. However, we believed that the relevance feedback algorithm would excel with negative expansion. To filter the results, it would not be necessary to find exactly those key terms that provide optimal filtering. In addition, relevance feedback is computationally less expensive and can provide a faster response.

Our second interest was in the nature of the best expansion technique for different users. Since other studies have indicated substantial variability in users' search styles and competence [Koenemann and Belkin 1996; Meyer et al. 1997; Nordlie 1999; Pitkow and Kehoe 1996; Thury 1998], we believed that the interaction between users with different competency levels and both the algorithms and the expansion type would differ. To find possible differences in algorithmic effects, we divided the user group into high, middle, and low achievers. We surmised that all groups would benefit from using the algorithms, but that high achievers would benefit more from negative expansion and low achievers would gain more from positive expansion. High achievers would be capable of formulating good queries themselves and might benefit from additional filtering. Low

achievers might need additional help in guiding the search by inclusion of more required terms.

## 5. USER STUDY

### 5.1 Design

To test our hypotheses, we designed a user study in which 30 subjects searched the Internet to find the answers to five broad questions, henceforth referred to as *topics*. The subjects were students in Management Information Systems. They received extra credit in one of their classes for participating in the experiment. Additionally, the best answer to each question would be rewarded with \$25.

The questions were taken from the TREC-6 conference (ad hoc topics, available from [http://trec.nist.gov/data/topics\\_eng/index.html](http://trec.nist.gov/data/topics_eng/index.html)). We selected broad topics that would require subjects to look for multiple Web pages for a complete answer. The following were the question themes: adverse effects of taking aspirin daily, advantages of dental implants, the parties involved and problems with fiber optic cable around the world, research into new fuel sources, and the achievements of the Hubble telescope. Subjects were asked to write down the Web pages that contained a complete answer, for example, a list of all achievements of the Hubble telescope, or a partial answer, for example, a single achievement of the Hubble telescope such as the discovery of Supernova 1987A. They were informed that we would evaluate the content of every Web page they wrote down and that the completeness of an answer would be decided based on the number of instances of information concerning the topic contained in the pages. A partial answer is henceforth referred to as an *instance*. The most complete answers would be rewarded. Subjects used our Java-interface so we could capture any clicking behavior, which revealed their implied attitudes about the value of a page among the results. This also allowed us to store the keywords used, the number of searches, the results of each search, and the URLs followed by the users for later evaluation. For each topic, the subjects had 10 min to find as many relevant Web pages as possible. Ten minutes is a time limit often used by TREC participants and found to be sufficient—for example, in a pure observational study by Hersh et al. [2001] most users required less than 10 min to finish a comparable task.

During preliminary testing, we first tested the algorithms with the AltaVista search engine ([www.altavista.com](http://www.altavista.com)). At that time, this engine returned the title together with the first lines of a document. The Google search engine ([www.google.com](http://www.google.com)) returns the text surrounding the keywords and we found that this provided better feedback to the algorithms. We therefore switched to Google. The interface connected to Google by sending a query formatted in the same manner as regular online queries. At that time, Google did not correct misspelled words. By using multiple threads, all queries from a population from the genetic algorithm were sent to Google simultaneously. Since there was a slight difference in processing time required by the algorithm, a time delay was added so that the results only showed after a few seconds in each condition and

subjects could not distinguish between conditions based on time required for processing. Subjects were informed about this.

We designed our study such that each subject took part in the five experimental conditions: a baseline with no algorithm active (None), relevance feedback with positive query expansion (RF+), relevance feedback with negative query expansion (RF-), genetic algorithm with positive query expansion (GA+), and genetic algorithm with negative query expansion (GA-). For each subject, the conditions were randomly assigned to the topics and the order of the topics was randomized. We assume that each topic is a fair representation of the general underlying search task we set our users, that is, to find information about a broad topic.

Subjects were post hoc divided into low-, middle-, and high-performance groups. Questionnaires customarily are used to divide subjects post hoc into groups or to adjust groupings based on scores such as domain knowledge, or demographic information such as gender or class standing [Kracker and Wang 2002; Specht and Kobsa 1999]. However, we wanted to divide subjects based on their actual and not their self-reported search expertise. We therefore did not use a pretest questionnaire but instead calculated the subject's overall performance as his or her average  $F$ -measure as defined by van Rijsbergen [1979]. We determined the subject's  $F$ -measure per condition and then calculated the average over all five conditions. Since this combined measure was independent of each particular condition's results, it could be used to divide the subjects without introducing bias. The high achievers would be the 10 subjects with the highest scores, the low achievers would be the 10 subjects with the lowest scores, and the middle achievers would be the 10 subjects with scores between the low and the high extremes.

## 5.2 Results

In the following we provide descriptive data on both user and algorithm searching, followed by precision and recall rates for the complete group, precision and recall per achievement group, and qualitative descriptive data for each group.

**5.2.1 Descriptive Data.** We designed the experiment for 30 users, but needed to withhold four subjects' data from our analyses because three did not attempt to solve all the topics and one searched for the answers to the dental implants topic during the aspirin session, that is, the subject switched to the keyword "Dentalimplant.com" during the aspirin topic. The 26 subjects performed on average 4.9 searches per topic. For the two methods using negative query expansion (RF- and GA-), there were slightly more searches: 5.3 and 5.1, respectively. Subjects used multiword phrases, single words, or a combination of the two to search. Counting both phrases and words as key terms, they used 1.9 terms per search. On average, one-third of the search terms were single words (37%) and two-thirds were phrases (63%). In many cases subjects seemed to "copy and paste" part of the actual question for the topic and submit this to the system as a search phrase. For an overview, see Table I.

Figure 1 provides an overview of the percentage of the subjects' searches that were modified by the algorithms. The "System-modified Queries" represent user

Table I. Overview of General Search Characteristics

Condition ( $n = 26$ )	Searches per topic	Key terms per search	Words/phrases per search (%)
None	4.5	1.9	34/66
RF+	4.4	1.7	36/64
RF-	5.3	1.9	38/62
GA+	4.9	2.0	40/60
GA-	5.1	1.9	39/61
Overall	4.9	1.9	37/63

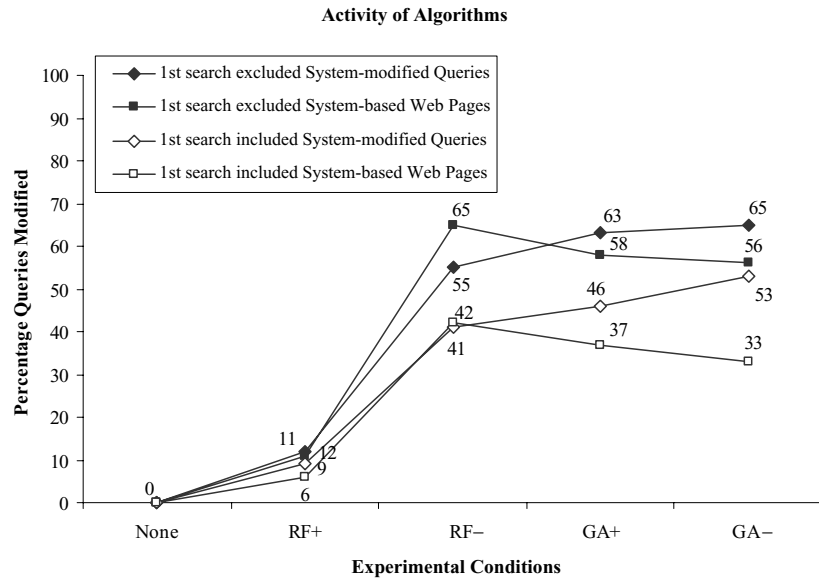


Fig. 1. Percentage query modification and resulting Web pages recorded by users.

queries modified by the algorithms resulting in more than five results. When we included the first search in the results, over 40% of the queries were modified during RF-; 37% and 33% were modified under GA+ and GA- conditions. However, since the algorithms had not yet received any feedback during the first search in a session, no modification was possible at this point. A more precise number that reflects the true algorithmic activity is the percentage of queries modified when modification was possible: starting with the second search. Using this correction, the algorithms modified more than half of the user queries. During RF- as many as 55% of user queries were modified. The effect of algorithm activity was significant,  $F(3, 100) = 14, 44$ ,  $p < .05$ , with a particularly significant jump from RF+ to RF- (Tukey pair wise comparisons,  $p < .05$ ).

Figure 1 also shows the system-based Web pages. These are the Web pages that contained the answer according to the users and that were retrieved by system-modified queries. As explained above, the more precise number is the one that excludes the first search. Relevance feedback for negative query expansion provided 65% of the Web pages that subjects recorded. An example of a modification by RF- is the original user query “Hubble Telescope,” which

was modified to “Hubble Telescope” but not “interactive, track, trec-6, title, trec-7.” For the genetic algorithm, 44% of recorded Web pages were based on a system-modified query. For example, GA+ modified a user query “Hubble Telescope launching” to the query “hubble telescope, hubble, achievements, accomplishments, telescope.” The number of recorded Web pages based on algorithm queries differed significantly between algorithms,  $F(3, 100) = 6.72, p < .05$ , and, as can be seen in Figure 1, the jump from RF+ to RF– was especially significant.

These results show that for the genetic algorithm proportionally fewer Web pages from modified queries were selected. For example, 65% of the queries were modified by GA–, which accounted for 56% of the Web pages. However, for RF– this was reversed; there were proportionally more pages selected from modified queries, that is, 65% of the recorded Web pages came from the modified queries (55%).

**5.2.2 Precision and Recall.** To evaluate precision and recall, two experienced librarians who hold master’s degrees in Library Science collaborated to compile a gold standard containing Web pages relevant to the five topics using Google. In addition, they evaluated all Web pages written down by users when searching for each topic. Relevant Web pages found by study participants were added to the gold standard. The content of each Web page was then further analyzed to compile a gold standard of instances that answered each topic. For the five topics related to “aspirin,” “dental implants,” “fiber optics,” “new fuel sources,” and “Hubble telescope” there were 45, 51, 32, 47, and 32 Web pages in the gold standard, which contained respectively 18, 20, 42, 56, and 58 instances of information. For example, for the Hubble telescope topic, one instance included in the gold standard was that the Hubble telescope provided evidence of supermassive black holes residing in the centers of many galaxies.

We calculated both precision and recall for the full group and for the three achievement groups. Precision was calculated for each topic as the number of relevant Web pages provided by the subject divided by the total number of Web pages provided by the subject. A Web page was relevant if it was included in the gold standard. Recall was calculated as the total number of instances found by the subject divided by the total number of instances in the gold standard. Although this measure of recall might favor longer pages, which probably contain more instances, this does not affect our evaluation since none of the algorithm parameters were based on or influenced by the length of a page. The distinction between Web pages for precision and instances for recall was necessary because finding more relevant pages does not necessarily mean that more instances are found. For example, a subject may have found five relevant pages, each containing a single achievement of the Hubble telescope, while another subject may have found one Web page containing, an exclusive list of all achievements of the Hubble telescope. Even though the first subject found five Web pages, more information was found by the second who only listed one Web page.

Table II. Precision and Recall of the User Answers

Condition	Precision (%)	Recall (%)
None	38	14
RF+	42	16
RF-	45	16
GA+	37	13
GA-	50	14
Overall	42	15

**5.2.2.1 Precision and Recall for the Complete Group.** Precision was highest when the genetic algorithm was used for negative query expansion (50%) and lowest when no algorithm was present (38%) or with the genetic algorithm used for positive query expansions (37%), as is shown in Table II. For recall, the five conditions were almost identical. Subjects retrieved on average 15% of the available information. In some cases recall was zero because the subject wrote down a single Web page that contained no relevant information. For example, during their search some subjects retrieved the TREC Web pages that contained the actual question they were asked to solve for the topic. This is a relevant link to follow since the description of the page displays the actual question. Unfortunately, the page itself contains only the question and not the answer to it. Even so, two subjects wrote down this Web page as containing the answer.

**5.2.2.2 Precision and Recall for the Three Achievement Groups.** In the next phase we divided the 26 subjects into three groups based on their average  $F$ -measure, as discussed above. The nine subjects with the highest average  $F$ -measures were designated as high achievers, the nine subjects with the lowest average  $F$ -measures constituted the low achievers, and the eight subjects in between became the middle achievers. The average  $F$ -measures were 51, 45, and 30 for the high, middle, and low achievers.

Figures 2 and 3 show how each group performed in precision and recall under the five conditions. The differences between the groups were significant for both precision,  $F(2, 127) = 5.07, p < .05$ , and recall,  $F(2, 127) = 13.59, p < .05$ . In general the genetic algorithm with negative expansion had the most profound impact on user performance. It had a beneficial effect on the performance of low achievers: precision doubled and recall tripled compared to other conditions. However it hindered high achievers. The following provides more detailed analysis per achievement group.

To compare the performances of the groups under the experimental conditions (RF+, RF-, GA+, GA-) against our baseline (None), we performed a one-way analysis of variance (ANOVA) for each group. Since we needed to drop four subjects, a two-way ANOVA (group  $\times$  experimental condition) would have been unbalanced. For the low achievers there was a significant main effect of the experimental condition on precision,  $F(4, 40) = 2.73, p < .05$ . In particular, the difference between GA+ (14% precision) and GA- (57% precision) was significant (Tukey pairwise comparisons,  $p < .05$ ). Although

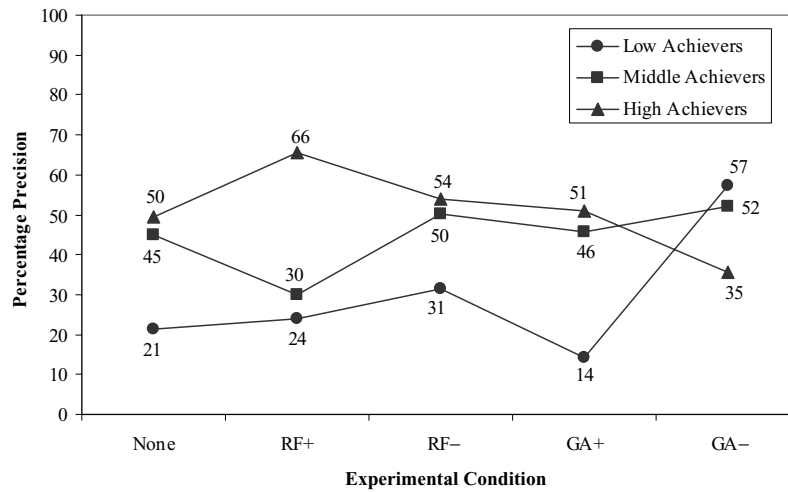


Fig. 2. Precision for the three achievement groups.

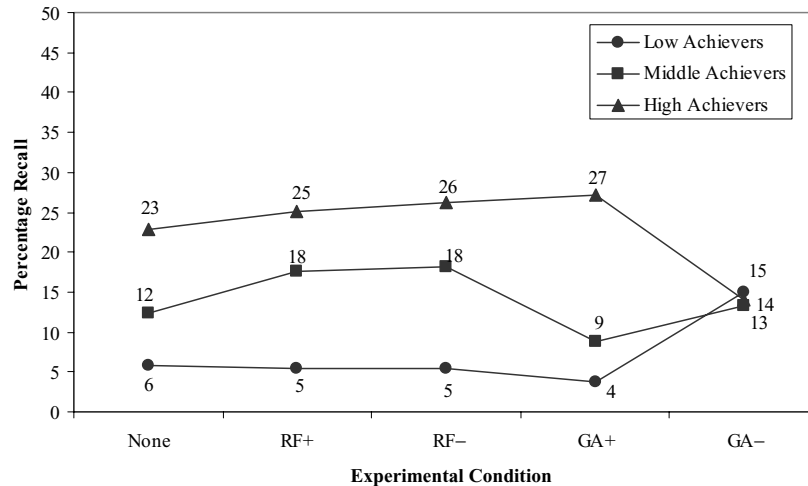


Fig. 3. Recall for the three achievement groups.

not significant, there was a strong trend ( $p = .117$ ) toward improved recall under GA-. For both the high achievers and middle achievers, the differences were not significant. The large difference in recall and precision between GA+ and GA- for high achievers was not significant due to variability in the data.

To compare the effects of the type of expansion on each algorithm, we performed two-way ANOVAS for each group and looked at the algorithms (RF vs. GA) and the expansion type (positive vs. negative expansion). For the low achievers there was a significant effect on precision of the algorithms used,

$F(1, 32) = 6.94$ ,  $p < .05$ , with the genetic algorithm demonstrating more precision. There was a strong trend ( $p = .07$ ) for an interaction between algorithms and expansion, indicating that, especially for the genetic algorithm, the difference between the two was significant. Similar effects were found for recall for the algorithms and the interaction between algorithms and expansion. In contrast, we found a trend among the high achievers to demonstrate higher precision with RF than with the GA ( $p = .09$ ). There were no significant differences in recall in this group. Additionally there were no significant differences in precision or recall for the middle achievers.

**5.2.2.3 Qualitative Characterization of the Achievement Groups and Algorithms.** To get better insight into the three groups and how the different algorithms affected them, we further looked at seven characteristics of subjects' behavior. We counted the number of searches per topic and the number of key terms used (single words or phrases). We also looked at the quality of the URLs followed by the subjects during the experiment. To evaluate these URLs, we asked the experts to score all URLs clicked on by the users with a 0 (irrelevant), 0.5 (probably relevant), or 1 (definitely relevant). This evaluation was done before any Web pages were reviewed and was based on information available in the descriptions that accompanied the links. Additionally, we counted the number of links followed by users to find an answer, the number of Web pages written down as containing the answer, and the number of searches having no results. Table III contains an overview of eight characteristics for each group.

High achievers seemed to use fewer key terms ("Key terms used per search") but used more phrases instead of single words ("Percentage phrases used per search"). They also made the least spelling errors. The middle achievers made twice as many spelling errors as the low achievers. They performed more searches and wrote down more Web pages ("URLs written down as answer per topic") per topic. Although the middle group followed the most links per topic ("URLs clicked per search"), the total quality and also the average quality of clicked URLs was highest for the high achievers. Finally, high achievers had far fewer searches (first or other) that could not retrieve documents ("Searches without results" listings).

The experimental conditions were randomized per user and the topics were randomly assigned to each condition. However, users might become more adept at using a search interface when using it with subsequent tasks. To evaluate such a possible effect, we calculated precision and recall for each consecutive task. Table IV provides an overview. Neither precision nor recall increased continually with each subsequent task for the complete group. Precision lowered with subsequent tasks for the complete group average but only by a small margin. The patterns differed for individual groups. Low achievers reached the highest precision while working on the third task, middle achievers while working on the second task, and high achievers while working on the first task. Recall varied slightly from task to task. Low achievers reached the highest recall while working on the fourth task, middle achievers with the third task, and high achievers with the fifth task.

Table III. User Characteristics for the Three Achievement Groups (Numbers are averages unless noted differently.)

	Low Achievers	Middle Achievers	High Achievers
Searches per topic:	4.7	4.5	5.4
Misspellings per search:	0.3	0.6	0.07
Key terms used per search:	1.9	2.2	1.6
Percentage phrases used per search:	62	52	73
Quality of URLs clicked per search:	2.0	2.1	2.9
Total quality of URLs clicked per search:	2.8	4.3	6.2
URLs clicked per search:	1.9	2.4	2.2
URLs written down as answer per topic:	3.6	3.4	5.0
Percentage of <i>first</i> searches without results due to user search terms	13	17	4
Percentage of <i>all</i> searches without results due to user search terms	10	7	7

Table IV. Precision and Recall Depending on Order of Task

	Task 1	Task 2	Task 3	Task 4	Task 5
Precision	%	%	%	%	%
Low Achievers:	15	24	46	31	31
Middle Achievers:	53	61	36	29	44
High Achievers:	67	35	47	59	46
All:	45	39	44	40	40
Recall					
Low Achievers:	2	3	10	12	8
Middle Achievers:	17	19	21	3	11
High Achievers:	26	14	20	22	33
All:	15	12	17	13	17

Figure 4 contains an overview of the subjects' and algorithms' contribution to the total group recall and shows that users and algorithms can provide complementary results. The gray area represents the percentage of recall contributed by users. This combines information found by all users without algorithmic help or found both with and without help. The black area represents the additional information found solely when algorithms were applied. Figure 4 shows that the genetic algorithm provided additional information to the low achievers although relevance feedback did not. The genetic algorithm doubled the number of relevant instances found by the group of low achievers. For both middle and high achievers there seems to exist a tradeoff, with users contributing less when the algorithms provided more information, resulting in a fairly consistent level of recall. For the high achievers, both the user and algorithm contributions were low under the GA- condition.

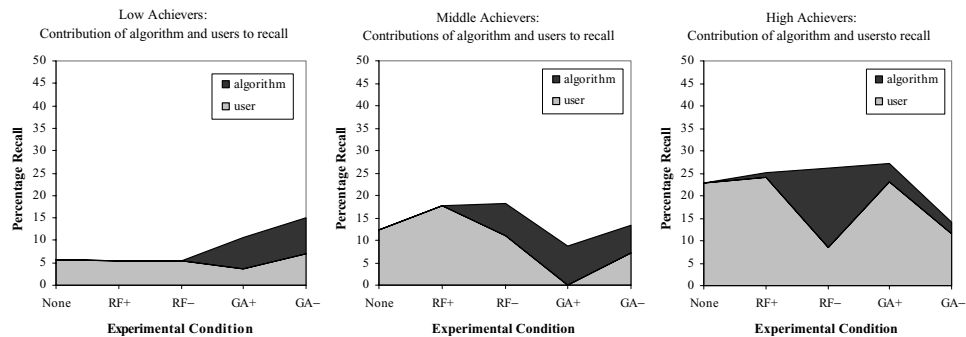


Fig. 4. Contribution to recall by users for three achievement groups.

## 6. DISCUSSION

We first want to address the limitations of our study. The results are based on a limited number of queries and users. However, since our study was based on actual users, a very large sample size of several hundred data points was impractical. We therefore reported on strong trends that might be significant with more users. Additionally, because of our use of an existing search engine, we needed to modify the relevance feedback implementation, which was here implemented with actual keywords, not with keyword weights. Our research was also based on broad search topics. Searching for the answer to a narrow question, which requires, for example, only a yes/no answer, might show different results. Since users searched for different aspects for each topic, they might have repeatedly retrieved the same documents during subsequent searches. This might have affected the querying speed and the clicking behavior of users but not the query expansion since it was sufficient feedback if a relevant link was clicked once. We assume that this limitation was comparable for all users in all conditions and therefore feel we can compare between the conditions.

We hypothesized that performance would increase, especially for the genetic algorithm conditions. Low and high achievers displayed opposing trends. The low achievers' performance was more than doubled with use of the genetic algorithm and negative query expansion. Under these conditions, they achieved better precision than the middle and high achievers in most conditions. The low achievers' recall when helped by the genetic algorithm equaled the middle achievers' recall, but stayed well below the recall attained by the high achievers. For the high and middle achievers there were no significant differences in performance, although the precision and recall graphs suggest that the genetic algorithm with negative expansion hindered the high achievers. In contrast to our expectations, the low achievers gained significantly from negative expansion, which provided an additional layer of information that was recalled without any tradeoff. With GA- help, this group outperformed both other groups. The most beneficial use of the algorithms would be to use them selectively for different groups. The genetic algorithm with negative expansion should be activated for low achievers but not for high achievers; relevance feedback should be activated for high achievers.

Our evaluation of recall was aspect oriented, that is, a higher recall was achieved when more instances for a certain topic were found. To maximize this type of recall, it was necessary to find related but different Web pages. Negative expansion was better for this than positive expansion. Users could use different keywords to address a different subset of the Web pages. The negative expansion filtered out nonrelevant documents without focusing the actual query on similar documents. Comparable, positive expansion might be expected to retrieve more similar documents, since the query would be composed of related terms, resulting in higher precision but lower recall. The results do not reflect this. Recall did not change much between different conditions and precision was not higher with positive expansion.

The lack of significant differences in recall for the full group is consistent with Thury's [1998] findings. She studied students' online search and browse behavior and noted that students seemed to keep on searching "*not until they found something worthwhile, but until an arbitrary trigger ended their search phase*" [emphasis added; Thury 1998, p. 267]. Our artificial cutoff point at 10 min might have strengthened this behavior. However, the subjective cutoff point seems different for high, middle, and low achievers. Users from both the middle and high achievement groups performed better than the low achievers. They might have decided that a larger number of Web pages would satisfy them. For both groups, a tradeoff between information retrieved by users and algorithms might have existed but recall did not change by much regardless of the performance of the algorithms or users, with the exception of the GA- condition for the high achievers.

To selectively help subgroups of users, it will be necessary to recognize to which group a subject belongs. We believe that no one factor, but a combination of factors, accounted for subjects belonging to a particular achievement group and governed their interaction with the algorithms. A major indicator seems to be the number of searches a user performs that do not result in any documents being found. Low achievers performed many searches without results. Furthermore, the low achievers used more single words than the high achievers and did not provide high-quality positive feedback. For example, one subject in the lowest achievement group searched for information on dental implants with the following queries: first, "dential implants" and "benefits" (notice the typo in the first term); then "fake teeth" and "benefits"; and, finally, "artificial teeth" and "benefits" (notice another typo in the first search term). The first and third queries did not return any results because of the misspelled words. The second query returned a list of results based on search terms of doubtful quality. This could explain why GA- was especially helpful to them: it increased precision without relying on extra user keywords or clicking behavior to guide the search. In contrast, the high achievement group used more specific queries and investigated higher-quality URLs, allowing higher-quality positive expansion.

More than half of the searches were modified by the algorithms. Analysis of the user data reveals that different reasons led the system to revert to the original user queries. One reason was our choice to add five keywords. Since these keywords were chosen from Web page descriptions that related to each

other and to the topic at hand, we expected these terms to be sufficiently related to make it possible to find Web pages containing them. This was not always the case. It would be interesting to add a variable number of keywords depending on the number of keywords already provided by the users. Lack of feedback and suitable keywords is a second reason. Sometimes subjects did not follow any links after the first search, so no positive feedback was available for positive query expansion. Furthermore, there were spelling errors, such as “acomplishments,” “reaseach,” “pedeatrics,” “enivorment,” “propsecting,” and long phrases, such as “Undersea Fiber Optic Cables” and “research on new fuel sources” that did not retrieve any results. It is a limitation of our algorithms that they could not explicitly eliminate these keywords and phrases from (continued) usage. Once part of a keyword pool, they were available to be reselected for query modification, resulting in many queries without results. A restart option (with an unmodified query) or a more advanced spelling correction system should be considered. However, in several instances the GA algorithm worked around this problem and replaced misspelled words—for example, the user query “Hubble” and “acvhiements” became “Hubble” and “telescope,” but not “images,” “esa,” “gallery,” “field,” or “project,” in the GA– condition. A last reason for algorithm queries without results was the lack of commonsense knowledge on the part of the algorithms. Both the RF– and GA– algorithms contained required and filtering query terms. Sometimes there was a contradiction between the two term sets, preventing the search engine from finding matching Web pages, for example, finding Web pages containing “dentistry” but not “dental” was often impossible.

## 7. CONCLUSION

This user study was performed with actual users in a realistic setting. Although the user group was fairly small, several unexpected, interesting results were found. Dividing subjects into different groups according to overall actual performance showed how each group interacted differently with the algorithms. The genetic algorithm significantly helped low achievers but seemed to hinder high achievers. In future implementations the difficulty will be correctly distinguishing the low achievers from others, although information available from subjects’ usage of single words, typos, etc., might provide clues. A comprehensive user study with more users might reveal stronger effects and might lead to the discovery of the clues to recognize users as high or low achievers based on their behavior.

A closer look at query expansion revealed an opportunity for improvements in the selection of potential keywords. Keywords that have no documents associated with them should be excluded from further inclusion by the algorithms. Additionally, some common sense should be added so a query does not contain contradictions arising from synonyms and filtering terms.

In general, we believe that the genetic algorithm is a promising technique for users searching the Internet. With additional improvements, genetic algorithms can greatly assist low achievers’ recall and precision. More research is needed to distinguish between the different achievement groups to examine the

potential impact of an improved genetic algorithm and relevance feedback for high achievers.

#### ACKNOWLEDGMENTS

We would like to thank the following people for their help, insightful comments, and suggestions: Jeffrey Scott from Arizona State University and Olivia R. Liu Sheng, Daniel D. Zeng, Michael Chau, and Paul B. Lowry from the University of Arizona. We also thank all our users for participating in our study and our reviewers for their time and thoughtful comments.

#### REFERENCES

- AMATI, G., CARPINETO, C., AND ROMANO, G. 2001. FUB at TREC-10 Web Track: A probabilistic framework for topic relevance term weighting. In *Proceedings of the Tenth Text REtrieval Conference (TREC 2001, Gaithersburg, MD)*. 182–192.
- ATTAR, R. AND FRAENKEL, A. S. 1977. Local feedback in full-text retrieval systems. *J. ACM* 24, 3, 397–417.
- BELKIN, N. J., COOL, C., HEAD, J., JENG, J., KELLY, D., LIN, S., LOBASH, L., PARK, S. Y., SAVAGE-KNEPSHIELD, P., AND SIKORA, C. 1999. Relevance feedback versus local context analysis as term suggestion devices: Rutgers' TREC-8 interactive track experience. In *Proceedings of the Eighth Text REtrieval Conference (TREC 8, Gaithersburg, MD)*. 565–573.
- BODNER, R. C. AND CHIGNELL, M. H. 1998. ClickIR: Text retrieval using a dynamic hypertext interface. In *Proceedings of the Seventh Text REtrieval Conference (TREC 7, Gaithersburg, MD)*. 573.
- BUDZIK, J. AND HAMMOND, K. J. 2000. User interactions with everyday applications as context for just-in-time information access. In *Proceedings of the 5th International Conference on Intelligent User Interfaces*. 44–51.
- CHEN, C. C., CHEN, M. C., AND SUN, Y. 2001. PVA. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 257–262.
- CHEN, H., CHUNG, Y.-M., AND RAMSEY, M. 1998a. A smart itty bitsy spider for the web. *J. Amer. Soc. Inform. Sci.* 49, 7, 604–618.
- CHEN, H., SHANKARANARAYANAN, G., AND SHE, L. 1998b. A machine learning approach to inductive query by examples: An experiment using relevance feedback, ID3, genetic algorithms, and simulated annealing. *J. Amer. Soc. Inform. Sci.* 49, 8, 693–705.
- CLAYPOOL, M., LE, P., WASED, M., AND BROWN, D. 2001. Implicit interest indicators. In *Proceedings of the International Conference on Intelligent User Interfaces (New York, NY)*. 33–40.
- DE LIMA, E. F. AND PEDERSEN, J. O. 1999. Phrase recognition and expansion for short, precision-biased queries based on a query log. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (Berkeley, CA)*. 145–152.
- FAN, W., GORDON, M. D., AND PATHAK, P. 2000. Personalization of search engine services for effective retrieval and knowledge management. In *Proceedings of the International Conference on Information Systems (ICIS, Brisbane, Australia)*. 20–34.
- FINKELSTEIN, L., GABRILOVICH, E., MATIAS, Y., RIVLIN, E., SOLAN, Z., WOLFMAN, G., AND RUPPIN, E. 2002. Placing search in context: The concept revisited. *ACM Trans. Inform. Syst.* 20, 1, 116–131.
- FULLER, R. AND DE GRAAFF, J. J. 1996. Measuring user motivation from server log files. In *Proceedings of the Conference on Designing for the Web: Empirical Studies (Microsoft Campus)*.
- HARMAN, D. 1988. Towards interactive query expansion. In *Proceedings of the Eleventh International Conference on Research & Development in Information Retrieval (New York, NY)*. 321–331.
- HARMAN, D. 1992. Relevance feedback revisited. In *Proceedings of the 15th International ACM/SIGIR Conference on Research and Development in Information Retrieval*.
- HAWKING, D. AND CRASWELL, N. 2001. Overview of the TREC-2001 Web Track (TREC 2001). In *Proceedings of the Tenth Text REtrieval Conference (Gaithersburg, MD)*. 61–68.

- HERSH, W., SACHEREK, L., AND OLSON, D. 2001. Observation of searchers: OHSU TREC 2001 interactive track. In *Proceedings of the Tenth Text REtrieval Conference (TREC 2001, Gaithersburg, MD)*. 434.
- HERSH, W., TURPIN, A., PRICE, S., CHAN, B., KRAMER, D., SACHEREK, L., AND OLSON, D. 2000a. Do batch and user evaluations give the same results? In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Athens, Greece).
- HERSH, W., TURPIN, A., SACHEREK, L., OLSON, D., PRICE, S., AND CHAN, B. 2000b. Further analysis of whether batch and user evaluations give the same results with a question-answering task. In *Proceedings of the Ninth Text Retrieval Conference (TREC 9, Gaithersburg, MD)*, 407.
- IDE, E. 1971. New experiments in relevance feedback. In *The SMART Retrieval System: Experiments in Automatic Document Processing*, G. Salton, Ed. Prentice-Hall, Englewood Cliffs, NJ, 337–354.
- IDE, E. AND SALTON, G. Interactive search strategies and dynamic file organization in information retrieval. In *The SMART Retrieval System: Experiments in Automatic Document Processing*, G. Salton, Ed. Prentice-Hall (1971), Englewood Cliffs, NJ, 373–393.
- JANSEN, B., SPINK, A., AND SARACEVIC, T. 2000. Real life, real users, and real needs: A study and analysis of user queries on the web. *Inform. Process. Manage.* 36, 2, 207–227.
- KOENEMANN, J. AND BELKIN, N. J. 1996. A case for interaction: A study of interactive information retrieval behavior and effectiveness. In *Proceedings of the Conference on Human Factors in Computing Systems* (Vancouver, B.C., Canada).
- KRACKER, J. AND WANG, P. 2002. Research anxiety and students' perceptions of research: An experiment. Part II. Content analysis of their writings on two experiences. *J. Amer. Soc. Inform. Sci. Tech.* 53, 4, 295–307.
- KRAFT, D. H., PETRY, F. E., BUCKLES, B. P., AND SADASIVAN, T. 1994. The use of genetic programming to build queries for information retrieval. In *Proceedings of the First IEEE Conference on Evolutionary Computation* (New York, NY). 468–473.
- LAI, H. AND YANG, T.-C. 2000. A system architecture for intelligent browsing on the Web. *Decis. Supp. Syst.* 28, 219–239.
- MAGENNIS, M. AND RIJSBERGEN, C. J. V. 1997. The potential and actual effectiveness of interactive query expansion. In *Proceedings of the the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 342–332.
- MEYER, B., SIT, R. A., SPAULDING, V. A., MEAD, S. E., AND WALKER, N. 1997. Age group differences in Word Wide Web navigation. In *Proceedings of the Conference on Human Factors in Computing Systems* (Atlanta, GA). 295–296.
- MICHALEWICZ, Z. 1992. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, New York, NY.
- NICK, Z. Z. AND THEMIS, P. 2001. Web search using a genetic algorithm. *IEEE Internet Comput.* 5, 2, 18–26.
- NORDLIE, R. 1999. User revelation—a comparison of initial queries and ensuing question development in online searching and in human reference interactions. In *Proceedings of the Twenty-Second Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Berkeley, CA). 11–18.
- PATHAK, P., GORDON, M., AND FAN, W. 2000. Effective information retrieval using genetic algorithms based matching functions adaptation. In *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*. 533–540.
- PITKOW, J. E. AND KEHOE, C. M. 1996. Emerging trends in the WWW user population. *Commun. ACM* 39, 6, 106–108.
- ROBERTSON, S. E. AND SPARCK JONES, K. 1976. Relevance weighting of search terms. *J. Amer. Soc. Inform. Sci.* 27, 3, 129–146.
- ROCCHIO, J. J. Relevance feedback in information retrieval. In *The SMART Retrieval System: Experiments in Automatic Document Processing*, G. Salton, Ed. Prentice Hall, Englewood Cliffs, NJ, 313–323.
- ROSS, N. C. M. AND WOLFRAM, D. 2000. End user searching on the internet: An analysis of term pair topics submitted to the excite search engine. *J. Amer. Soc. Inform. Sci.* 51, 10, 949–958.
- SALTON, G. AND BUCKLEY, C. 1990. Improving retrieval performance by relevance feedback. *J. Amer. Soc. Inform. Sci.* 41, 4, 288–297.

- SPECHT, M. AND KOBSA, A. 1999. Interaction of domain expertise and interface design in adaptive educational hypermedia. In *Proceedings of the Second Workshop on Adaptive Systems and User Modeling on the World Wide Web at the Eighth International World Wide Web Conference* (Toronto, Ont., Canada). 89–93.
- SPINK, A. 1996. Multiple search sessions model of end-user behavior: An exploratory study. *J. Amer. Soc. Inform. Sci.* 47, 8, 603–609.
- SPINK, A., WOLFRAM, D., JANSEN, M. B. J., AND SARACEVIC, T. 2001. Searching the Web: The public and their queries. *J. Amer. Soc. Inform. Sci. Tech.* 52, 3, 226–234.
- SULLIVAN, D. 2000. NPD search and portal site study. Search Engine Watch: <http://www.searchenginewatch.com/reports/npd.html>.
- THURY, E. M. 1998. Analysis of student web browsing behavior: Implications for designing and evaluating Web sites. In *Proceedings of the Sixteenth Annual International Conference on Computer Documentation* (Quebec, Canada). 265–270.
- TOMS, E. G., W. KOPAK, R., BARTLETT, J., AND FREUND, L. 2001. Selecting versus describing: A preliminary analysis of the efficacy of categories in exploring the Web. In *Proceedings of the Tenth Text REtrieval Conference (TREC 2001, Gaithersburg, MD)*.
- VAN RIJSBERGEN, C. J. 1979. *Information Retrieval*, 2nd ed. Butterworths, London, U. K.
- VOGT, C. C. 2000. Passive feedback collection—an attempt to debunk the myth of clickthroughs. In *Proceedings of the Ninth Text REtrieval Conference (TREC 9, Gaithersburg, MD)*. 141.
- WHITE, R. W., JOSE, J. M., AND RUTHVEN, I. 2001. Comparing explicit and implicit feedback techniques for Web retrieval: TREC-10 interactive track report. In *Proceedings of the Tenth Text REtrieval Conference (TREC 2001, Gaithersburg, MD)*.
- WHITE, R. W., RUTHVEN, I., AND JOSE, J. M. 2002. Finding relevant documents using top ranking sentences: An evaluation of two alternative schemes. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (Finland)*. 57–64.
- XU, J. AND CROFT, W. B. 1996. Query expansion using local and global document analysis. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. 4–11.
- XU, J. AND CROFT, W. B. 2000. Improving the effectiveness of information retrieval with local context analysis. *ACM Trans. Inform. Syst.* 18, 1, 79–112.
- YANG, J. J. AND KORFHAGE, R. 1993. Query optimization in information retrieval using genetic algorithms. In *Proceedings of the Fifth International Conference on Genetic Algorithms*. 603–611.
- YANG, K. AND MAGLAUGHLIN, K. 1999. IRIS at TREC-8. In *Proceedings of the Eighth Text REtrieval Conference (TREC 8, Gaithersburg, MD)*. 645.
- YANG, K., MAGLAUGHLIN, K., MEHO, L., AND SUMNER, R. G., JR. 1998. IRIS at TREC-7. In *Proceedings of the Seventh Text REtrieval Conference (TREC 7, Gaithersburg, MD)*. 555.

Received July 2002; revised January 2003; accepted May 2003