PERGAMON

Information Processing and Management 37 (2001) 789–816

INFORMATION PROCESSING & MANAGEMENT

www.elsevier.com/locate/infoproman

# Information navigation on the web by clustering and summarizing query results

Dmitri G. Roussinov [a,*], Hsinchun Chen [b,1]

[a] *School of Information Studies, Syracuse University, 4-234 Center for Science and Technology, Syracuse, NY 13244-4100, USA*

[b] *Department of MIS, College of BPA, University of Arizona, Tucson, AZ 85721, USA*

Received 10 July 2000; accepted 19 October 2000

**Abstract**

We report our experience with a novel approach to interactive information seeking that is grounded in the idea of summarizing query results through automated document clustering. We went through a complete system development and evaluation cycle: designing the algorithms and interface for our prototype, implementing them and testing with human users. Our prototype acted as an intermediate layer between the user and a commercial Internet search engine (AltaVista), thus allowing searches of the significant portion of World Wide Web. In our final evaluation, we processed data from 36 users and concluded that our prototype improved search performance over using the same search engine (AltaVista) directly. We also analyzed effects of various related demographic and task related parameters. 

*Keywords:* Information retrieval; Neural networks; Clustering; Summarization; Relevance feedback; World Wide Web; Internet spiders; Search engines

## 1. Introduction

In the 1980s and 1990s the cost of storing information electronically went down considerably, so available collections of text and multimedia documents proliferated. The Internet and the World Wide Web have made such collections readily available to the public. Communication technologies have contributed millions of electronic messages and sites representing a variety of

* Corresponding author. Tel.: +1-315-443-1892; fax: +1-315-443-5806.

*E-mail addresses:* droussin@syr.edu (D.G. Roussinov), hchen@bpa.arizona.edu (H. Chen).

[1] Tel.: +520-621-2748; fax: 520-621-2433.

0306-4573/01/$ - see front matter 
PII: S0306-4573(00)00062-5

interests. Personal home pages and business web sites have in recent years been accruing at a rate of hundreds of thousands a day. Our productivity in generating information has exceeded our ability to process it, and the dream of creating an information-rich society has become a nightmare of information overload.

In the realm of keyword searching, we need to match words in the documents being sought with the words that we enter into the system. For example, finding the answer to the question "How long does it take to get by train from Copenhagen to Oslo?" might seem possible by composing a query "copenhagen AND oslo AND train" in Boolean syntax. Entered into AltaVista (www.altavista.com), one of the most popular Internet search engines, this query results in about 900 matching web pages, only one of which contains the answer. It has been repeatedly noted in popular magazines (www.searchenginewatch.com) that users tolerate exploring only the first 20–30 pages in the ranked lists returned by search engines and then give up.

In addition, when users approach an information access system (Web or non-Web) they often have only a fuzzy understanding of how to achieve their goals. It is not surprising that iterative clarification is required while interacting with a computerized information access system (Belkin, Oddy, & Brooks, 1982), especially since a user's notion of relevance may change in the process of information seeking.

Although we believe that the findings reported here apply to many collections of comparable size, in undertaking the study we were particularly interested in Web searching, which many researchers and practitioners consider to be one of the most challenging and important areas for future research involving National Information Infrastructure (NII) applications (Bowman, 1994; DeBra & Post, 1994; Pinkerton, 1994). "Science Daily Magazine" wrote about the web (Lawrence & Giles, 1999):

> The web is transforming society, and the search engines are an important part of the process. Consumers use search engines to locate and buy goods or to research many decisions (such as choosing a vacation destination, medical treatment or election vote). Search engine indexing and ranking may have economic, social, political, and scientific effects. For example, indexing and ranking of online stores can substantially affect economic viability; delayed indexing of scientific research can lead to the duplication of work or slower progress; and delayed or biased indexing may affect social or political decisions.

Hearst (1999) reviewed a number of interactive tools that may potentially be effective for facilitating user information access, including popular and intuitively appealing approaches that are based on clustering and summarizing of search results. The idea behind them is to agglomerate similar documents into clusters and present a high-level summary of each cluster so the user does not need to peruse the full length of documents. Examples of recently developed systems based on summarizing query results are Scatter/Gather (Hearst & Pederson, 1996), Lexical Navigation (Cooper & Byrd, 1997), WebBook (Card, Robertson, & York, 1996), and SenseMaker (Wang Baldonado & Winograd, 1997). Hearst (1999) noted that no empirical evidence has been put forward to demonstrate that any of these tools make information access easier than the traditional keyword approach, while some negative results have been reported.

In our study, we first speculated on why prior experience with clustering/summarization may have shown negative results. We present our reasoning at the end of the "literature review"

section. Based on our conjectures and prior studies, we recommend several possible improvements and have designed an approach called *Adaptive Search*, which is based on a novel use of *clustering, summarization and user feedback*, explained in the "adaptive search implementation" section of this paper. Adaptive search allows users to find information in the significant portion of World Wide Web. It acts as a layer between a user and a traditional query-based search engine (AltaVista in this particular study). In order to test whether this layer improves search performance, we ran a controlled experiment and compared subjects' performance using our approach against using the search engine directly. Along with differences in other metrics, we found that users were able to find information sooner using adaptive search and succeeded in a higher proportion of tasks. This has addressed our research question: *whether there exists a combination of user feedback with automated summarizing that improves information-seeking performance* (increases productivity in searching).

In addition, we observed and explored user behavior while utilizing both approaches and analyzed our findings in an effort to determine the relative effectiveness of each for particular kinds of tasks and for particular kinds of users. Because we desired to maximize the effect of using adaptive search in place of traditional keyword searching (and also had limited resources for empirical exploration) we based our implementation on a combination of features (clustering, summarization, relevance feedback) rather than separately testing the effect introduced by each feature. We conjectured that testing each feature separately would result in effects so small that it would be almost impossible to distinguish them in so diverse and dynamic a domain as Internet searching.

The next section provides a review of the related literature. The section "adaptive search implementation" describes our approach in more detail. Then we describe our empirical study, followed by our conclusions.

## 2. Literature review

Research in interactive information seeking has produced many systems based on a number of innovative ideas, including those grounded on automated summarization and relevance feedback. We enlarge here only upon systems that have been empirically evaluated. We refer the reader to Hearst (1999) for a more comprehensive review.

### *2.1. Clustering and summarizing query results*

Scatter/Gather developed by Xerox PARC (Cutting, Karger, Pedersen, & Tukey, 1992) clusters documents into groups, and presents descriptive textual summaries of those groups to the user. The summaries consist of topical terms that characterize each cluster generally, and a set of typical titles that hint at the contents of the cluster. The user may select a subset of clusters that seem to be of interest and re-cluster their contents, thus examining the contents of each sub-collection at a finer granularity.

By running experiments with subjects, Pirolli, Schank, Hearst, and Diehl (1996) found that application of Scatter/Gather to an entire test collection indeed communicated a high-level picture of a collection of documents. They asked subjects to perform specified search tasks and then to

describe a hierarchy of concepts into which they would identify items in the collection. The subjects who used Scatter/Gather were in much closer agreement about their understanding of topics and also provided richer descriptions of the collection than did subjects who used keyword searches. However, more time was required to perform the search tasks with Scatter/Gather and the documents found were judged less relevant. The study concluded that search based on exploration of a hierarchy of clusters was not superior to the keyword-based approach but suggested that a combination of clustering and keyword search might be superior to both. Exploring such a combination was among the motivations for our study.

Hearst and Pederson (1996) integrated Scatter/Gather with conventional search technology by clustering the results of a query. They found that relevant documents tended to fall mainly into one or two out of five clusters, and that the precision and recall were higher within the best cluster than within the retrieval results as a whole. On average, 60–100% of the relevant documents appeared in the "best cluster" and approximately 80–100% in the best two. The implication of this finding is that by filtering out all clusters of irrelevant documents one time, a user might save significant time by looking only at the contents of the seemingly most promising clusters.

However promising these findings, an empirical study of overall search effectiveness of their implementation has not yet been conducted. Hearst (1999) also noted that

> An intriguing extension to this idea is to use the output of clustering of retrieval results as the input to a relevance feedback mechanism, either by having the user or the system select the cluster to be used, but this idea has not yet been evaluated.

We at least partially followed this suggestion in designing our approach.

### 2.2. Relevance feedback

The concept of feedback originated in early work in the field of cybernetics (Wiener, 1948). Spink (1997) identified several types of feedback used in information retrieval, including *document relevance feedback* and *term relevance feedback*. She also noted that studies of relevance feedback have been largely confined to "consideration of users' post-interaction judgments," and are "descriptive" in nature.

*Document relevance feedback* is an interaction cycle in which the user identifies certain retrieved documents as relevant, and the system then retrieves another set of documents, trying to take the user's judgment into consideration. Document Relevance feedback has been found to be an effective mechanism for improving retrieval in a number of studies (Salton & Buckley, 1990; Buckley, Salton, & Allan, 1994).

However, in "real life" information retrieval, such as Web searching, users rarely use features based on document relevance feedback. An example is the "more like this" feature in the Excite search engine evaluated by Jansen, Spink, Bateman, and Saracevic (1998). One possible explanation is that in order to make relevance judgments the user still has to read through the documents, which may be time consuming. Although it has been demonstrated that document relevance feedback indeed helps searchers to progress toward their goals over a sequence of iterations, we are not aware of any empirical study that has shown that a search with document relevance feedback is more efficient than the manual query reformulation approach. To us,

efficiency is the level of overall performance measured as search time, or number of query reformulation, or number of documents, or other performance measures, similar to those used in our study.

*Term relevance feedback* (TRF) is a type of feedback in which the user identifies a term (or terms) within the retrieved items and subsequently uses those terms to modify the query (Spink, 1997). Koenemann and Belkin (1996) compared several types of term relevance feedback:

*Control Group*. No relevance feedback allowed. The subjects could only reformulate the query by editing.

*Opaque*. Subjects simply selected relevant documents, thus providing document relevance feedback, although they could not see how the system reformulated the queries.

*Transparent*. Same as opaque but, in addition, the subjects could see which terms the system added.

*Penetrable*. Subjects were also allowed to select or deselect terms to be added from a list shown in transparent type.

The outcome of the study indicated that the search times did not differ significantly among the conditions, but the differences in the number of feedback iterations were significant, with the subjects in the penetrable group being the most efficient, followed by the opaque group. The opaque group required more cycles than first two. The transparent group's performance was found to be worse than those of either the control or opaque group. All subjects preferred relevance feedback to the baseline system, considering it a 'lazy' approach to selecting suggested terms instead of thinking up their own.

The term relevance feedback approach is similar to another approach called *interactive query expansion* (Magennis & van Rijsbergen, 1997), in which "the potential query expansion terms are shown to the searcher as suggestions. The searcher then decides which to add and which not to add." Magennis and van Rijsbergen (1997) noted that none *of the implementations of* [*either*] approach has been empirically found to lead to improved retrieval effectiveness. Harman (1988) simulated a "perfect" choice by the user and found an improvement in a small scale Cranfield 1400 collection. In Magennis and van Rijsbergen (1997) simulation of "experienced" users and a large scale TREC collection revealed improvement, but an experiment with five human subjects did not. They used 20 queries suggested by TREC conference (http://trec.nist.gov/), 4 queries per subject.

A number of approaches have been based on user selection of additional search terms suggested by a system from manually or automatically created *thesauri* (Smeaton & van Rijisbergen, 1983; Croft & Das, 1990; Chen & Schatz, 1993). These techniques typically use existing indexing vocabularies but either translate user searching terms into indexing terms or provide a list of alternative searching terms that are related to the user's input searching term. Our approach in this study used no thesaurus, thus avoiding significant manual or computational expenses associated with creating one.

### 2.3. Web search studies

Internet is a relatively new medium, able to communicate previously unseen magnitudes of information and characterized by a number of unique peculiarities. Jansen et al. (1998) explained the growing interest Web information seeking research by pointing out that "While Internet

search engines are based on IR principles, Internet searching is very different from IR searching as traditionally practiced and researched,'' and that at there currently exists limited knowledge of user searching behavior.

A number of studies have evaluated retrieval performance of the best known of currently available commercial web search engines (Chu & Rosenthal, 1996; Gauch & Guijun, 1996). Ding and Marchionini (1996) studied the precision of the top 20 retrieved pages by using only five queries. Westera (1996) also used five queries, all dealing with the topic of wine.These researchers composed their own queries, utilizing their own expertise and mastery of Boolean query language by, for example, manually adding synonyms to enhance the queries.

Our study went a few steps further. We explored subjects' ability to interact with the system and find desired information. We believe that this component is crucial for evaluation, since many users have difficulty composing Boolean queries (Young & Shneiderman, 1993). Furthermore, when used by non experts, using expert queries do not reveal an engine's performance.

A panel on World Wide Web searching during the ACM Conference on Research and Development in Information Retrieval (SIGIR 98) may be considered a first attempt to formalize evaluation of the ability to find information on the web. Panel participants received 10 tasks by email several months before the conference. These tasks were mostly related to tourism/entertainment topics such as ''What does it cost to ride on the upper deck of the Star Ferry across Hong Kong harbor to Tsimshatsui?'' We list all the tasks in Table 3 when we discuss our results. Then, the participants tried using a search engine of their choice to find the answers. It took between 1 and 40 min for each task to be performed. An answer to every question was found by at least one participant. AltaVista search engine (used in our study for benchmark comparisons and as a component of our system) turned out to be the most popular among the participants.

A very popular Information Retrieval community, TREC conference (Voorhees & Harman, 1999), has recently adopted a track for Web search. An 18.5-million-page snapshot of part of the Web (frozen at a specific time) has been created to enable reproducible evaluation of Web search systems and techniques. The collection comes with a set of tasks and web pages identified as relevant to those tasks. Our study differs from TREC along a number of dimensions. First, we used a significantly larger portion of the Web for empirical evaluation, in particular the entire part indexed to AltaVista, which was estimated by http://www.searchenginewatch.com/ at 250 million pages at the time of our experiment (January–May 1999). Second, we incorporated users' information seeking into our evaluation process, whereas TREC queries have been traditionally composed by researchers themselves.

### *2.4. Clustering web documents*

Recently, researchers have suggested applying clustering associated in order to facilitate information search. Mechkour, Harper, and Muresan (1998) speculated that seeking information on the Web often is difficult because of the great heterogeneity of available documents. They suggested ''intelligent assistance to filter the information available on the WWW and reduce the scope of view to specialized subsets... to the area of interest of each particular user or group of users.'' Their system, called WebCluster, was based on clustering a smaller collection of documents (called a *source* collection), thus ''organizing the documents into classes of closely related documents.'' subsequently using those classes in order to automatically compose more efficient

queries for commercial search engines, thus hopefully resulting in more precise results. They also noted that clustering the entire WWW content is not a feasible task and that the research prototype must rely on using commercial search engines as its retrieval mechanism (back end). Thus, their system was implemented as an intermediate layer between the user and Internet search engines (similar to the one our system reported in this study).

Other systems based on Web document clustering features have been suggested (Zamir, Etzioni, Madani, & Karp, 1997; Sahami, Yusufali, & Baldonado, 1998). NothernLight commercial search engine (http://www.northernlight.com/) organizes retrieved pages into folders. However, no user studies incorporating those features have so far been reported.

### 2.5. Kohonen's self-organizing maps

Our clustering and summarizing approaches are based on Kohonen's self-organizing maps (SOM) (Kohonen, 1995). Developed in the 1980s, SOM has since been successfully used for dimension reduction and clustering as well as in many applications such as pattern recognition and natural language processing. Chen, Schuffels, and Orwig (1996) showed that SOM could efficiently summarize a collection of text documents. Orwig, Chen, and Nunamaker (1997) showed that SOM could identify key concepts mentioned in a collection of text documents.

As an unsupervised neural network, SOM is known to be more resistant to typically noisy input (Chen, 1994) characteristic of statistical clustering techniques such as single-link or K-means (Jain & Dubes, 1988).

Kohonen's SOM is an unsupervised neural network. It does not need examples for training as supervised neural networks do. SOM can accept objects described by their features as input and place those objects on a 2D map in such a way that similar objects are placed close together. In this sense it is similar to multidimensional scaling (Jain & Dubes, 1988).

The topology of the Kohonen SOM network is shown in Fig. 1. Each output node corresponds with a node in a 2D grid. The network is fully connected in that every mapping node is connected to every input node with some connection weight. During the training phase, the inputs are presented several times to train the connection weights in such a way that distribution of output

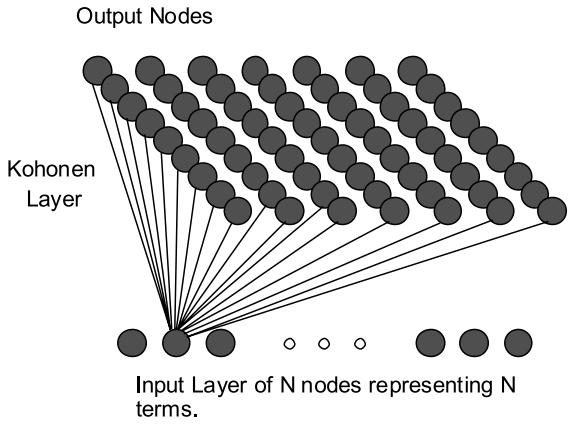


Fig. 1. Kohonen SOM network topology.

nodes represents distribution of input points. The network trains fully automatically, without any human intervention.

An outline of a revised SOM algorithm for textual classification is summarized below, primarily to demonstrate that it is very simple to implement. More details can be found in (Orwig et al., 1997).

(1) *Initialize input nodes, output nodes, and connection weights*: Create a 2D map (grid) of $M$ output nodes (say a 20-by-10 map of 200 nodes). Initialize weights $w_{ij}$ from $V$ input nodes to $M$ output nodes to small random values. This way, each input node corresponds to a coordinate axis in the document vector space. Each output node is associated with a vector of weights $w_{ij}$, so it can also be considered a point in the input vector space.

(2) *Present each document in order*: Describe each document as an input vector $x_i(t)$ of $V$ coordinates. Set a coordinate to 1 if the document has the corresponding term and to 0 if there is no such term. A term can be a single word or a phrase. The vector size $V$ typically is set in the 100–1000 range and only the top-most frequently occurring $V$ words or phrases are used as terms. Each document is presented to the system several times.

(3) *Compute distance to all nodes*: Compute Euclidean distance $d_j$ between the input vector at time $t$, $x_i(t)$, and each vector of weights $w_{ij}$ representing an output node:

$$d_j = \sum_{i=0}^{N-1} (x_i(t) - w_{ij}(t))^2.$$

(4) *Select winning node $j^*$ and update weights to node$j^*$ and its neighbors*: Select winning node $j^*$, which produces minimum $d_j$. Update weights to nodes $j^*$ and its neighbors to reduce the distances between them and the input vector $x_i$ $(t)$:

$$w_{ij}(t+1) = w_{ij}(t) + \eta(t)(x_i(t) - w_{ij}(t)).$$

After such updates, the nodes in the neighborhood of $j^*$ become more similar to the input vector $x_i(t)$. Here, $\eta$ (t) is an error-adjusting coefficient $(0 < \eta(t) < 1)$ that decreases over time. Please see Kohonen (1995) for the algorithmic details of neighborhood selection and adjustment.

(5) *Label regions in map*: After the network is trained through repeated presentations of all documents (each document is presented at least five times), assign a term to each output node by choosing the one corresponding to the largest weight (*winning term*). Neighboring nodes that contain the same winning terms are merged to form a region (cluster). Assign each input document to the node with the closest vector of weights $w_{ij}$. Consequently, the resulting map represents regions of concepts with the documents assigned to them. Concept regions that are similar (conceptually) automatically appear in the same neighborhood.

In our experiments, building the map for 200 pages took only 3–6 s on a DEC Alpha 3000/600 workstation (200 MHz, 128 MBs RAM).

### 2.6. Justifying our approach

This section explains our rationale behind the implementation used in our study. The next section provides more details on the architecture of adaptive search.

*2.6.1. Rank order instead of a hierarchy*

Studies have shown that users easily become disoriented while navigating hierarchies, both manually created (such as Yahoo) or automatically generated (such as self-organizing maps) (Chen et al., 1996). Navigating a hierarchy requires retaining in memory all potentially interesting paths in the hierarchy that the user did not pursue but might consider returning to if the information sought had not been found. Navigating may be time-consuming and frustrating if the desired information is deep in the hierarchy or not available at all. On the other hand, presenting documents in rank order has become standard for modern search systems, including Internet search engines. We conjectured that we could reduce cognitive load by avoiding hierarchical exploration and instead decided to *present the documents to the user only by rank ordered lists.* This way, our presentation differs from the one supported by the Nothernlight search engine (http://www.northernlight.com/), where users look for pages by inspecting folders of their choice.

*2.6.2. Only term relevance feedback*

Since it has been observed that providing document relevance feedback is time-consuming and not popular among real searchers (Jansen et al., 1998), we decided to implement only *term relevance feedback*. Its not requiring user judgments about document relevance is the principal way in our implementation differs from Koenemann and Belkin (1996). In our implementation, the user may provide feedback on the terms that appear in the summary. Section "adaptive search implementation" provides more details and examples.

*2.6.3. Allow three types of term relevance feedback*

Since prior studies have shown that clustering conveys a high-level picture of document collections (Pirolli et al., 1996), we conjectured that a user might be able to provide feedback to the system after taking a brief look at the clusters of query results. Adaptive search allows three major types of feedback: (1) selecting what is relevant, (2) rejecting what is not relevant, and (3) specifying what is missing. "Adaptive search implementation" section provides the details.

*2.6.4. How our summarization works*

Many interactive systems that are based on relevance feedback provide only the first two types of feedback listed in the previous paragraph. However, we believe that type 3 (specifying what is missing) is also crucial for success. In order to support it, we need to create some sort of a summary of search results, ideally as comprehensive as possible. Since prior studies indicated that Kohonen's SOM can provide summarization of collections of text documents (Chen et al., 1996), we chose it as our way of summarizing. In the prototype used in our study, we limited representation of a self-organizing map to those allowed by HTML mark-up language, thus discarding graphical representations used in prior studies (Roussinov & Ramsey, 1998), an example of which is shown in Fig. 2. The section titled "adaptive search implementation" provides more details on our HTML representation.

The following explains why clusters of terms may offer a better summary than a simple list of most-frequent terms. SOM clusters text documents and displays sets of clusters, each described by the most representative terms inside that cluster. Imagine for example, a collection generated by a query "saturn" that has two major topics: a topic about cars and a topic about planets. It is possible that the topic about cars has many more documents than the topic about planets, so the

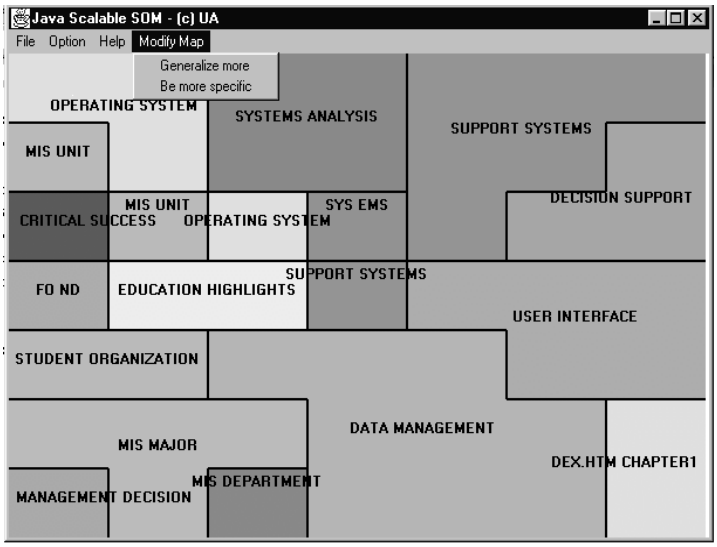


Fig. 2. An example of graphical representation of Kohonen's self-organizing map of the top 200 web pages retrieved by AltaVista search engine as a result of a query "MIS grants." Each region represents a cluster of pages with similar content. From Roussinov and Ramsey (1998).

most frequently used terms are likely to appear there. In this case, the user would not even see terms about the planets in the list of most frequent terms and might assume that none were in the query results. Such a list of most-frequent terms would not be an appropriate summary. On the contrary, if clustering has been applied, the user probably would see two clusters of different sizes, one with terms about cars and one with terms about stars.

## 3. Adaptive search implementation

Our implementation of adaptive search reported in (Roussinov, Tolle, Ramsey, & Chen, 1999), from which we have borrowed some figures for illustrative purposes, is described here in more detail.

### *3.1. Commercial search engines capabilities*

We created our prototype (adaptive search) as a layer between a user and a commercial Internet search engine. Almost any Internet search engine could be used as a basis for adaptive search. This section explains the features of an Internet search engine on which adaptive search relies. Because it would have been cost prohibitive to index all (or significant portion of) pages in the World Wide Web, we had to accommodate the reality that only a leading commercial search engine can afford the amount of traffic and index storage space associated with such an undertaking.

We decided on using AltaVista as one of the most popular and most comprehensive engines, according to evaluations done by popular magazines (www.searchenginewatch.com). It is worth

noting that we did not find answers to any of the search tasks used in our study in the pages listed in the Yahoo Internet directory (www.yahoo.com) at the time of the experiment (January–May 1999).

Most commercial Internet search engines accept queries in the form of text strings, composed according to rules (syntax) specific to each engine. The engine makes a guess about the user's interests and returns a list of documents, ordered by their perceived relevance. Most search engines, including AltaVista, Google, Lycos, Infoseek, Excite, and Hotbot, support the following features:

*Feature* 1. *Ability to specify what words or phrases should influence rank order*. For example, users of AltaVista "Simple Search" simply enter those words or phrases in the query line. For example, the query "hong kong tsimshatsui" resulted in the placing of documents containing words "hong", "kong" and "tsimshatsui" closer to the beginning in the rank ordered list of found pages. More occurrences of those words on a page led to a higher chance for the page to be at the beginning of the list. Search engines also use metadata to influence rank ordering. For example, the presence of certain words in the title would rank a document higher.

*Feature* 2. *Ability to require certain words to be present in the found pages*. In the AltaVista Simple Search, this is done by placing a "+" sign right before the word or phrase. For example "+hong +kong tsimshatsui" would find only documents containing "hong" and "kong" together. This is equivalent to the "AND" operator in many Boolean query languages: "hong AND kong." In Simple Search, words preceded by "+" sign also affect rank order, as explained in the preceding paragraph. So, the engine not only returns pages containing all of those words, but also orders them such that pages containing many occurrences would appear earlier.

*Feature* 3. *Ability to exclude documents containing certain words* . In AltaVista Simple Search, this is done by placing a "−" sign right before the word or phrase in the query. For example "+hong +kong tsimshatsui −view" would never return any pages containing the word "view."

*Feature* 4. *Ability to return the number of indexed web pages containing the specified words or phrases*. This feature is intended to help users refine their queries. This number's being very large indicates that the word is too general and probably not very useful. The number's being very small or zero usually indicates a spelling error or other reason for the word's infrequent appearance on the web.

### 3.2. Adaptive search paradigm

Interactions between the user, the system and the search engine are shown in Fig. 3. They proceed through the following steps:

*Step* 1. The user submits to the system a simple text description of the information need. No query language is required. The system sends the same description to a commercial search engine (SE), thus relying on feature 1 of the search engine discussed in the previous subsection. The search engine returns a ranked-order list of documents matching the query.

*Step* 2. The system fetches (from the Web) the 200 top documents in the ordered list and builds a self-organizing map for them (Kohonen, 1995). The self-organizing map algorithm automatically clusters documents and assigns a labels to each cluster. Using a so called *scalable self-organizing map algorithm* (Roussinov & Chen, 1998) allows building each map in less than a minute on a DEC Alpha 3000/600 workstation (200 MHz, 128 MBs RAM).

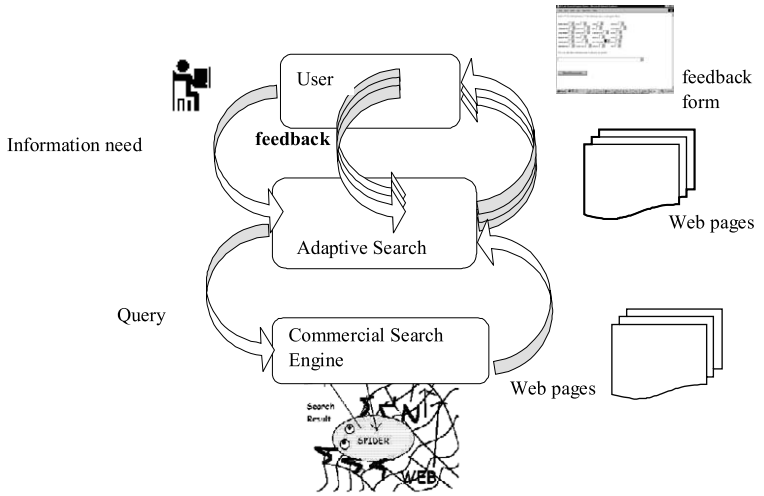


Fig. 3. Adaptive search approach. (*Source*: Roussinov et al., 1999.)

Based on the information contained in the map, the system generates and presents to the user a summary in HTML form. The next section provides more details about this form and the SOM algorithm.

*Step* 3. The user marks labels and terms on the summary form according to their relevance to the current information need and may type additional words or phrases describing the information believed to be missing in the summary. The sections below provide more details.

*Step* 4. Based on the selected and entered terms the system creates a sequence of queries and sends them to the search engine. More details are presented in the following sub-sections.

*Step* 5. The user browses the rank-ordered list of documents and inspects the pages that seem promising, based on snippets that the commercial search engine provides.

*Step* 6. Iterate. The user may fill out the feedback form differently and re-submit it, thus going through steps 3–5 repeatedly in order to find the pages of interest.

### *3.3. Summary and feedback form*

The summary/feedback form shown in Fig. 4 is a simplified textual representation of a Kohonen's SOM. As has been explained, SOM creates clusters of documents (called *regions*), each labeled with a unique word or phrase. One line of the form describes one region (cluster) in the self-organizing map. The first word or phrase (in boldface) labels the region. The next three most representative terms for the region follow the label term.

Users give feedback by marking words or phrases as "close to" or "far from" the information need, using a "+" or a "−" sign from a pull-down menu. Initially, we used checkboxes, but later opted for giving the user to have an opportunity to stay "neutral" with respect to a term by default, giving the user had three choices, each supported by a pull-down menu item.

The descriptive term list ($6 \times 4 = 24$ in the example in Fig. 4) double-functions as a document summary: the user can add any missing key topics on the bottom line.

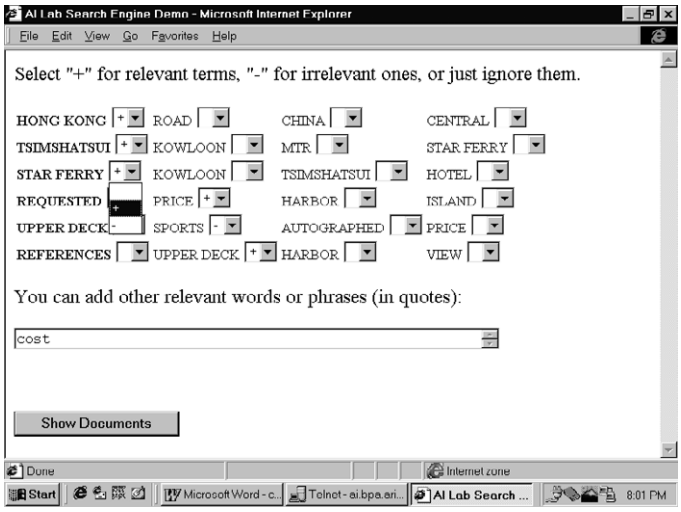


Fig. 4. An example of an HTML form generated by AS for the Star Ferry task. (*Source*: Roussinov et al., 1999.)

### *3.4. How feedback was used*

After a number of preliminary studies, we eventually settled on a heuristic algorithm that gave the best results.

The general idea behind the algorithm was first to create a so-called ''ideal query'' by combining all user feedback. If the ideal query returned too few documents (10 or less for the first cut in our current implementation) the algorithm modified this ideal query in order to get at least 20 (or more if requested by the user) documents. Since our component was on a different server from the underlying search engine, we had to minimize the amount of communication between them. Although our algorithm may not be optimal from the point of view of precision, it does guarantee real-time interaction.

The algorithm constructed an ideal query such that the matching documents:

1. Would have all the words the user marked as close to the information need or entered on the additional line. To achieve this, the algorithm used Feature 2 described in the preceding section.
2. Would not have any of the words the user marked as ''far from information need.''(Using Feature 3 in the preceding section.)
3. Would be rank-ordered according to the words marked as ''close to the information need'', words entered on the additional line or entered at the very beginning (step 1). (Feature 1 in the preceding section.)

The following example helps to clarify this algorithm. At the very beginning (step 1) the user typed the sentence ''what does it cost to ride on the upper deck of the Star Ferry across Hong Kong harbor to Tsimshatsui.'' Adaptive search created and displayed the HTML form shown in Fig. 4. The user marked ''hong kong'', ''tsimshatsui'', ''upper deck'', ''price'', and ''star ferry as close to the information need''. The user also marked the word ''sports'' as ''far from information need,''and typed the word ''cost'' in the additional input line. The ideal query expressed in AltaVista syntax was:

Table 1
Document frequencies on the web for the terms related to Star Ferry task

| Term | Number of pages on the Web containing the term |
|---|---|
| Hong kong | 1, 534, 031 |
| Tsimshatsui | 2,578 |
| Upper deck | 69, 226 |
| Price | 18,241,027 |
| Star Ferry | 593 |
| Sports | 10, 428, 227 |

`What does it cost to ride on the upper deck of the Star Ferry across Hong Kong harbor to Tsimshatsui + "hong kong" + tsimshatsui + "upper deck" + price + "star ferry" + cost - sports`

According to the search engine syntax, this required matching documents to have the words `hong kong`, `tsimshatsui`, `upper deck`, `price`, and `star ferry`. The matching documents would not have the word `sports`. Also, the presence of any of the words `what`, `does`, `it`, `cost`, `to`, `ride`, `on`, `the`, `upper`, `deck`, `of`, `star`, `ferry`, `across`, `hong`, `kong`, `harbor`, `hong kong`, `tsimshatsui`, `upper deck`, `price`, `star ferry`, or `cost` would make the rank of matching documents higher.

To obtain more matching documents, the algorithm modified the ideal query by removing from the query the required (prefixed with "+" in the ideal query) or excluded (prefixed with "–") words/phrases that also were very frequent. The frequencies of occurrence of a word or a phrase throughout the web (called *document frequencies*) were determined by querying the search engine (Feature 4 in the preceding section). Table 1 shows the document frequencies for the required terms from the example above.

The table indicates that the phrase `star ferry` occurs very rarely, while the word `price` occurs quite often. The algorithm therefore would have dropped the term `price` before `star ferry`. If several terms had to be dropped, the algorithm would have followed a heuristic strategy based on the widely used "inverse document frequency" weighting (Salton & McGill, 1983). The strategy of our algorithm was to maximize the objective function, which was the sum of $\log(N/\mathrm{d}f_i)$ for each term $i$ remaining in the query. Here, $\mathrm{d}f_i$ was the number of web pages containing the term $i$ (document frequency) and $N$ was the maximum $\mathrm{d}f_i$ among all the terms in the "ideal" query: $N = \max\{\mathrm{d}f_i\}$, for $i = 1$ to $M$, where $M$ was the number of required words and phrases in the ideal query. This ensured that the logarithm above always existed. This strategy would have discarded the most frequently occurring words, since they would have had smaller weights.

## 4. Empirical evaluation

This section describes an empirical study comparing the adaptive search (AS) approach and direct use of the Internet search engine, which we denote as a query based search (QBS). 36 undergraduate students in the business school agreed to be subjects in this study. Table 2 summarizes their demographic characteristics.

Table 2
Subjects demographic characteristics

| Number of subjects | 36 |
|---|---|
| Ratio of males/females | 25/11 |
| Proportions of age groups (20−/20−24/25−30/40+) accordingly | 6/22/4/4 |
| Self reported search engine use (daily/weekly/biweekly/monthly/less frequently) accordingly | 15/16/4/1/0 |
| Number of native speakers of English | 27 |

We accumulated data gradually over time, subject by subject. We initially planned to involve around 40 subjects, but had to cancel some sessions due to heavy network traffic. We have already reported intermediate results based on the data obtained from the first 24 subjects in our SIGIR poster (Roussinov et al., 1999) but present here a more detailed analysis and several new metrics, such as virtual task performance, user time (described below). Although we obtained statistically significant results based on the first 24 subjects, we decided to proceed with the initially planned 36 subjects in order to obtain more convincing results.

Based on the number of metrics, our subjects using *AS scored 10%–30% better than those using QBS*. We based our metrics on the time that it took the subjects to accomplish the assigned tasks, number of pages visited, the proportion of successful tasks, and the overall usefulness of the returned pages. Below, we first describe and justify each metric in more detail, formulate our hypothesis in terms of those metrics and provide an analysis of our data.

### *4.1. Experimental design and assumptions*

We assumed search performance to be a function of the tool (QBS or AS), a particular subject, and a particular search task. The tasks were randomly assigned. Each task was performed the same number of times with each tool. In our preliminary studies we had observed significant variation in subject skills. Therefore we decided to expose subjects to both interfaces instead of more traditionally splitting them into treatment/control groups, since the latter could have resulted in performance variation solely due to a difference in average group skill. Instead, half of our subjects used AS first, and the other half used QBS first. We assumed that we could ignore carry-over effect from one interactive search task to another. Later, we found no statistically significant difference between subjects' performance while working on their first task and second tasks, validating our assumption.

The subjects performed search tasks proposed by the panel on Web Search at the 1998 ACM Conference on Advances in Information Retrieval (listed in Table 4). The search tasks were very specific and clearly defined, mostly related to entertainment/tourism topics like: "What does it cost to ride on the upper deck of the Star Ferry across Hong Kong harbor to Tsimshatsui?" The tasks represented "needle in a haystack" questions to which only 2–10 pages on the web were expected to contain the answer. Since the tasks were very unambiguous, it was easy for the supervisor to decide whether the answer had been found.

Our subjects performed two kinds of search tasks, "interactive" and "virtual." The interactive search tasks required the subject to find a web document containing an answer to the search question. Since these tasks were very time-consuming, each subject was assigned to

perform only two of them, randomly assigned from a total set of 10 search tasks used in the study.

The remaining eight virtual search tasks required subjects to submit their queries (in the case of QBS) or the feedback form (in case of AS) only once. The less time-consuming virtual tasks provided more data. Although they did not permit direct comparison of the two systems, they allowed comparison of user performance at the first cycle of interaction. Since the subsequent interaction cycles were similar, we believed it reasonable to assume that the differences in performance in them would be also similar. That is, we assumed performance in the first interaction cycle to be indicative of overall performance.

### *4.2. Metrics*

Beyond the traditionally used in Information Science precision and recall (Salton & McGill, 1983), prior research has produced many useful metrics for evaluating interactive information access systems. Such alternative metrics include time required to learn the system, time required to achieve goals on benchmark tasks, and error rates (Hearst, 1999).

#### *4.2.1. Physical and user time*

We measured the time it took to find the answer to a given question. We assumed that a better system requires less time for information seeking. For the reasons explained below, we selected two metrics based on the time: physical time and user time.

We analyzed our log files after pilot experiments and came to the conclusion that the AS system itself usually spent considerably more time processing user inputs than QBS, 95% of that time being spent on communication with AltaVista. This time could have been reduced or eliminated if AS had been located on the same server as AltaVista. We concluded that more accurate metrics would use only the time a user spends searching (*user search time*), disregarding waiting time. Waiting time is a function of underlying technology and could easily be improved and, because cognitive load during waiting is not as high as during active interaction, summing the two may be misleading. The user search time was little influenced by the network transmission times, which was convenient for the experiment.

These considerations explain our choice of user search time as our primary metric. Our system automatically recorded in its log files the times when each user received an output from the system as well as the time a user submitted his/her input. Thus, we were able to compute the time that the system spent processing (including the major amount devoted to communication with AltaVista on a different server), which we later subtracted from the *physical time* to obtain the *user time*. Thus, the user time included time that a user spent browsing the search results and providing feedback, but not the We analyzed our log files time spent waiting for the system. This approach assumes that thinking about the task while waiting did not significantly influence user performance.

By following the experiment design, we allowed subjects to spend up to 20 min (physical time) searching. In our statistical analysis we treated the unsuccessful tasks same way as the tasks that accomplished at exactly 20 min, thus faced censored distribution with a ceiling (Cohen, 1959). We present in this paper both results reflecting our correction for censoring and obtained when ignoring the effects of censoring (since they are most easily interpreted as approximately average

search times). We used the reciprocal transformation $f(x) = 1/x$ as a normalizing correction. After the normalization the results were less sensitive to the cut-off because the tails had less influence on the sample mean. For example, the difference between 1/20 and 1/30 is much smaller in absolute terms than the difference between 20 and 30, so the exact cut-off number (20 or 30) is much less important.

Similarly to using 20 min as a threshold for the physical time, we established a threshold for the user time. However, since we obtained user time from the log file, we faced a slightly more complicated situation. We had two choices: (1) specify a threshold prior to the experiment and (2) obtain the threshold from the experiment data. Because the first approach required a good estimate of the user time, which we did not have, we decided to follow the second approach and to compute the threshold after the experiment, based on its data. We chose to use as the threshold the minimum user time for tasks for which the subject did not find the answer.

We refer to this time as $t^*$ and suggest a simple intuitive explanation for why, during the design phase, we believed it would be a good threshold time.

Each subject spent at least $t^*$. Those for whom the system was working faster due to less web traffic or other factors had an opportunity to spend more than $t^*$ (up to the 20 min in a hypothetical case when the processing time was negligible. Setting the threshold lower than $t^*$ would treat more accomplished tasks as unaccomplished and thus discard additional data. Setting the threshold time greater than $t^*$ would result in unfair comparison across subjects since not all of them had a chance to spend $t^*$ searching.

Fig. 5 illustrates this reasoning by displaying a histogram of the maximum time a user was able to spend, computed as 20 min minus processing time. This time ranged from 13 min (the longest processing time about 7 min) to almost 20 min (the shortest processing time almost 0).

From the experiment data we found $t^*$ to be 13.27 min. Thus, while statistically analyzing user time, we considered all tasks that required more than 13.27 min of user time as non-accomplished and assigned them 13.27 (the threshold) min. Later we confirmed that our statistical results were not substantially sensitive to the threshold selection.

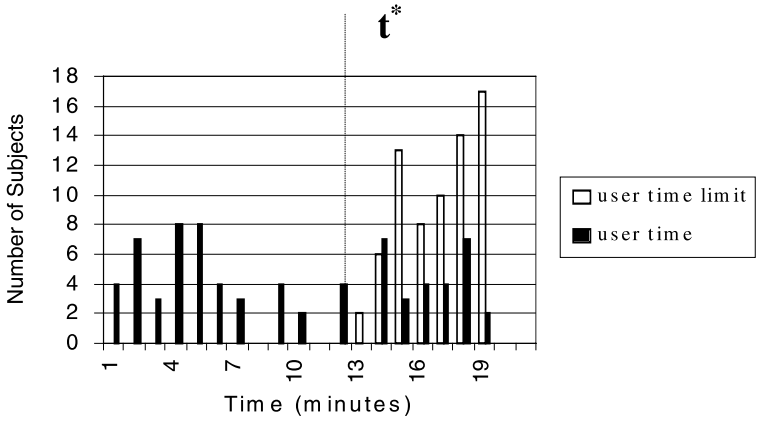


Fig. 5. Illustration of user time threshold: hystograms of user time and user time limit. Each subject was able to spend at least $t^* = 13.27$ min.

#### 4.2.2. Answer rank

To evaluate the quality of the returned rank-ordered lists of documents, we used another metric. We analyzed rank-ordered lists returned by both systems when subjects performed virtual tasks to find the first page containing the answer to the question contained in the task. We called the position of this page in the list *Answer Rank*. Ideally, Answer Rank should be 1. This metric is less direct than the one based on time, but is more stable since it depends on fewer random factors in the experiment, such as web traffic and subject ability to comprehend documents.

We chose this metric rather than the traditional Information Science notion of *precision* (Salton & McGill, 1983) because of our main purpose. We were interested in having the subject find at least one document containing the answer to the search task We were not interested in composing "efficient" queries that matched as many answers as possible. Precision and recall measures have been widely used for comparing the ranking results of non-interactive systems, but are less appropriate for assessing interactive systems (Lagergren & Over, 1998).

We also analyzed the number of pages that subjects visited in order to find the answers, including the pages with search results returned by the systems (AS and QBS). We analyzed only the cases where subjects found the answers.

#### 4.2.3. Other metrics

### 4.3. Hypothesis

Our null hypotheses are listed below:

H1: It takes the same user time to do the tasks with either tool.

H2: It takes the same physical time to do the tasks with either tool.

H3: It requires visiting the same number of pages $t$ in order to find the answer.

H4: Using both tools results in the same Answer Rank.

The alternative hypotheses were that AS performed better according to the metrics described above. We also tested the hypotheses H1*, H2* and H4*, which were the same as H1, H2 and H4 but applied to the data normalized by the reciprocal function $f(x) = 1/x$.

### 4.4. Procedure

The experimental procedure with each subject was the following:

1. fill out pre-questionnaire (5 min),
2. perform tutorial (10 min),
3. perform first task with interface 1 (QBS or AS) (up to 20 min physical time),
4. perform second task with interface 2 (AS or QBS) (up to 20 min physical time),
5. perform 8 virtual tasks while switching interfaces (approximately 10 min),
6. fill out post-questionnaire (5 min).

The textual description of a search task served as the starting point for both interfaces. The supervisor gave the task description to the subject. If the subject used adaptive search, he/she entered the description into the system (step 1), received the HTML form from the system (step 2), and provided feedback by marking words or phrases as "close to" or "far from" the information

need. Then the subject submitted the form and received an ordered list of documents, which he/she explored to find the answer. The subjects were allowed to make repeated form submissions (step 6).

If the subject used QBS, he/she was free to enter a query using AltaVista Simple Search syntax, which also allows entering the text description of the task unaltered. Entering entire text description never resulted in finding an answer. Very few subjects actually followed that strategy.

The tutorial consisted of explaining AltaVista's simple query syntax and search strategy. The supervisor explained the notion of rank order, use of the "back button" in the web browser, and the "find inside a page" functionality. Each subject was asked to find the answer to the question "What is the capital of Honduras?" Then, the adaptive search approach was explained, using the same tutorial task. The supervisor spent approximately the same amount of time for the tutorial using each interface and followed the same script with each subject.

All user actions such as buttons pressed, queries entered and pull-down menu selected were automatically recorded by the server's CGI script along with all the web pages visited by the subjects. The supervisor recorded timing.

### 4.5. Example of an interactive search session

This section presents an example of an interactive search session. The subject typed in the question "What does it cost to ride on the upper deck of the Star Ferry across Hong Kong harbor to Tsimshatsui?" and received the form shown in Fig. 4. As it turned out, "Upper Deck" is also the name of a company that makes sports equipment, which explains why some sports- related concepts appeared in the summary. The subject marked with "+" the concepts: `hong kong`, `tsimshatsui`, `upper deck`, `price`, `star ferry`; and with the "−" the concept `sport`. After pressing the "Show Documents" button, the user received the list of ranked documents shown in Fig. 6. The second web page contained the passage shown in Fig. 7, which clearly contains the answer to the search question.

Another subject who was asked to use a query based search form entered: "`hong kong`" + "`Star Ferry`".

This query resulted in a list of documents that were mostly about sightseeing from the Star Ferry and did not mention either the cost of a ticket on the upper deck or Tsimshatsuias the destination. An example of a more efficient query would be: +"`hong kong`" + `tsimshatsui` + "`star ferry`" + "`upper deck`" + `cost`.

Very few subjects were able to compose a query resembling this one. The potential pitfalls already have been identified in the literature, for example by (Young & Shneiderman, 1993): forgetting keywords, mismatched vocabularies, wrong use of syntax, too short or too long queries that result in too many or no documents.

### 4.6. Results and discussion

Table 3 displays several statistical results for interactive tasks that are discussed below in this section. We have to reject null hypothesis 1. This result provides strong evidence that AS requires less time, despite high involvement of random factors such as differences in web transmission times, subject skills, visiting fewer pages to find answers using AS. The reciprocal (normalized)

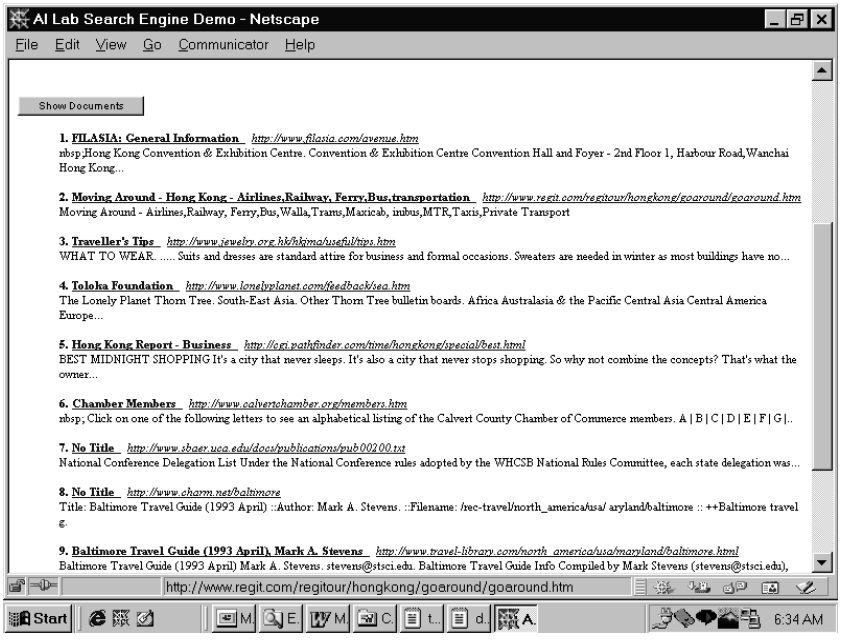


Fig. 6. A list of documents found by AS for the Star Ferry question. (*Source*: Roussinov et al., 1999.)

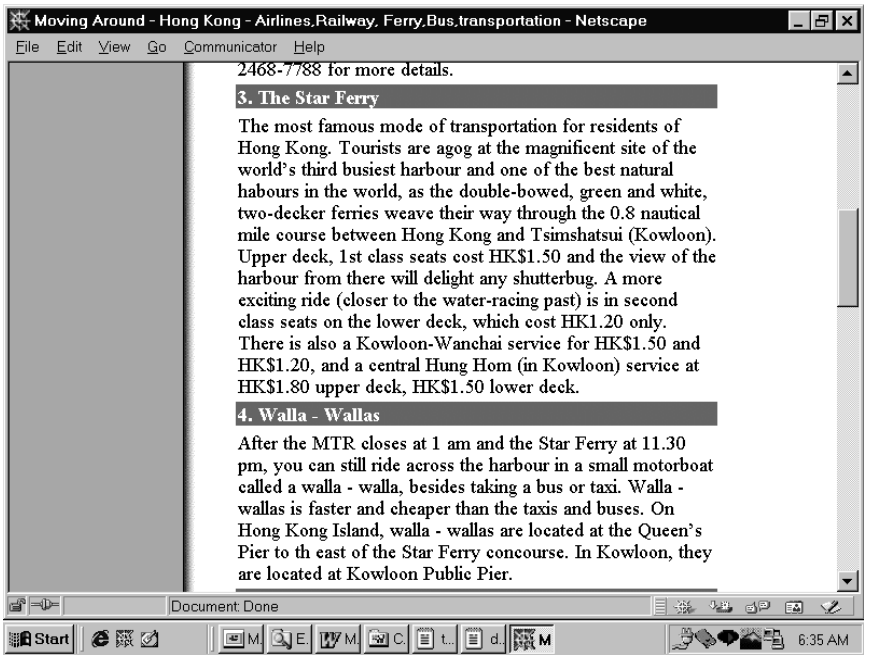


Fig. 7. A passage from a web page containing the answer to the Star Ferry question.

Table 3
Interactive tasks statistical results: adaptive search vs query based search

| Metric average | Tasks used | QBS | AS | Hypothesis | *t*-test, *p*-value |
|---|---|---|---|---|---|
| User time | Real | 8.7 min | 7.3 min | Reject H1 | 0.08 |
| Physical time | Real | 13.1 min | 11.9 min | Cannot reject H2 | 0.19 |
| Reciprocal user time | Real | 0.21 | 0.28 | Reject H1* | 0.04 |
| Reciprocal physical time | Real | 0.13 | 0.16 | Cannot reject H2 * | 0.18 |
| Number of pages visited | Real | 5.07 | 4.17 | Reject H3 | 0.090 |
| AR | Virtual | 12.86 | 10.60 | Reject H4 | 0.002 |
| Reciprocal AR | Virtual | 0.03 | 0.04 | Reject H4* | 0.01 |
| Proportion of tasks accomplished | Real | 15/36 | 21/36 | | |

metrics resulted in the same conclusions, with the *p*-value even more convincing for the user time *t*-test (0.04).

We observed that subjects spent approximately 95% of their time reading web documents or searching result pages, and only approximately 5% on typing or making menu selection. Thus, difference in typing demands while working with different interfaces is extremely unlikely to explain the above result.

We realize that *p*-values for our null hypothesis 1, 2, 3 are relatively high, so the findings solely based on interactive tasks are not completely convincing. Our statistical findings based on virtual tasks provided much lower *p*-values, making them more reliable. Other metrics, such as number of pages visited and proportion of tasks accomplished, also support our conclusion.

We had decided to check only whether the top 19 pages contained answers to the given question. We assigned Answer Rank (AR) of 20 to those lists that did not contain the answers in any of the top 19 pages. As a result, in order to obtain AR, we analyzed about $40 \times 10 \times 20 = 8000$ pages, including some duplicates. This analysis required a total of approximately 80 person-hours. The person responsible for analyzing pages did not know from which system a particular page had come, so there was no bias toward either system. We rejected hypothesis 4 and concluded that AS consistently positioned the user closer to finding the answer than QBS.

We observed that many subjects incorrectly rejected some of the pages containing answers after looking at the snippets (summaries) provided by the search engines. Also, the subjects often missed an answer in a web page containing it. These two facts may have contributed to reducing the time difference, even when AS consistently provided better lists of matching documents.

As responses to the questionnaire indicated, most of the subjects preferred AS to QBS. We asked them to choose an integer between 1 and 5 to describe their degree of preference between the two tools, 1 corresponding to the strongest preference of QBS and 5 to the strongest preference of AS. The average subject preference turned out to be 3.6, which is more statistically significant than 3 (corresponding to indifference between those two tools) with *t*-test *p*-value $=$ 0.003. Some reasons subjects gave for their preference for AS over QBS included: they did not need to come up with keywords on their own. They did not need to know query syntax. They had an HTML form with a set of concepts (created after entering a textual description of the task) as a starting point for their search instead of a blank input line, as in the case of QBS.

We also ran similar tests using reciprocal AR. This way the results should be much less sensitive to the cut-off (19 pages in our case) because the tails have less influence on the sample mean. For example, the difference between 1/20 and 1/30 is much smaller in absolute terms than the difference between 20 and 30, so the exact cut-off number is much less important.

To address concern about using repeated measures (collected from the same subject in different tasks), we also analyzed data subject by subject. We computed the "effect" for each subject as the difference in average reciprocal answer ranks:

$$\text{effect} = \text{average RAR using AS} - \text{average RAR using QBS}.$$

The effects were independent. They are displayed in Fig. 8. The mean effect (0.12) was positive ($t$-test $p$-value $= 0.013$). Positive average effect testifies to the superiority of AS in this experiment.

Notably, the average AR achieved by the subjects using QBS after their second query submission was also inferior to the one achieved by the very first submission of the feedback form by subjects using AS: 14.5 vs 10.2 (only "interactive" sessions were considered), $t$-test $p$-value $= 0.0026$. Thus, even after looking at the rank-ordered list and sometimes at several documents at the top of that list, the subjects were not able to modify their queries to achieve the same relevance of query results as subjects using AS attained after a single submission of the HTML form.

For analysis purposes in seeking to find out for which types of users this effect would be stronger, we divided subjects into two groups of equal size according to their average performance with both tools. We found that the effect was stronger for the less successful searchers. The means were significantly different, with $t$-test $p$-value $= 0.016$. This suggests that the adaptive search approach is likely to be more effective for novice users, those who usually are less efficient while searching the Web.

We also tried to predict performance by the level of the class from which a subject was recruited (from an introductory class to a 3rd year core class). Again, we distinguished two groups of subjects (numbering 17 and 14; 5 subjects were excluded since we did not have enough data about them) of presumably "more skilled" and "less skilled" subjects. We indeed observed average performance in the more skilled as superior (0.35 vs. 0.27), measured by average reciprocal AR ($t$-test $p$-value $= 0.02$). However, the effect was also higher in the more skilled group ($t$-test $p$-value $= 0.02$), which seemingly contradicts the finding reported in the preceding paragraph. We concluded that class level may not be a reliable predictor of the overall performance and the effect of a given search method.

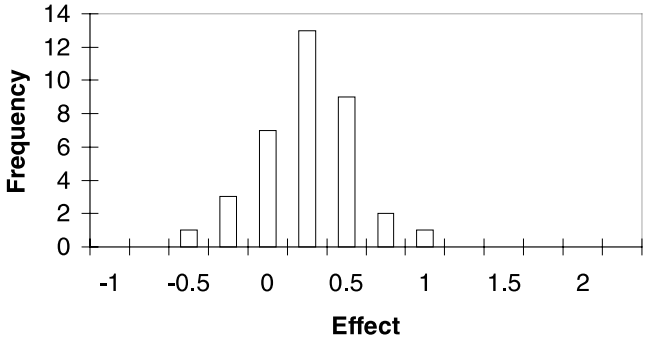


Fig. 8. Histogram of the effect by subjects.

Table 4 shows average AR for all 10 tasks. AR were significantly better for adaptive search in tasks 2, 3, 9, 10; worse for task 4 and not statistically different for tasks 1, 5, 7. AR were the same and invariably equaled 20 for tasks 6 and 8, since no subjects were able to obtain answers in the top 19 pages for those tasks with either interface.

The overall conclusion still held when we removed one task at a time and repeatedly re-ran our analysis. This indicates our results are not sensitive to task selection.

After pretests, we hypothesized that the effect would be stronger for the tasks that seemed more difficult in the sense that very few pages on the web contain answers to them. We divided tasks into two groups of four according to the number of answer-pages found through pretests and the experiment, and ignored tasks 6 and 8 since no subjects found answers for them.

For each subject, we computed the difference in effects between those measured by the ''tough'' tasks (1, 3, 9, 10) and by the ''easy'' tasks (2,4,5,7). The mean difference was found to be

Table 4
Average AR for each virtual task

| Task # | Task description | Average AR | | *t*-test, *p*-value | AS is | |
|---|---|---|---|---|---|---|
| | | Query based | Adaptive | | Better | Worse |
| 1 | I want to find where Max Beerbohm, the English caricaturist, lived in at the end of his life. | 12.38 | 13.29 | 0.34 | | |
| 2 | What does it cost to ride on the upper deck of the Star Ferry across Hong Kong harbor to Tsimshatsui? | 10.69 | 2.91 | 0.0004 | √ | |
| 3 | Where can I get a good pfeffersteak in Hagerstown, MD USA? | 13.21 | 6.94 | 0.013 | √ | |
| 4 | If I visit Singapore, what, if any, buildings designed by I. M. Pei can I see there? | 6.16 | 11.56 | 0.007 | | √ |
| 5 | Names of hotels in Kyoto (Japan) that are near the train station? | 15.81 | 15.29 | 0.39 | | |
| 6 | What is the cost of overnight train tickets, including sleeper accommodations (double occupancy) from Paris to Munich? | 20 | 20 | | | |
| 7 | How long does it take to get by train from Copenhagen to Oslo? | 10.73 | 9.88 | 0.36 | | |
| 8 | Was the Ring Cycle performed at Bayreuth, Germany, in summer 1998? | 20 | 20 | | | |
| 9 | I'm looking for the names of campgrounds around Lake Louise (Canada) that have showers. | 16.94 | 10.61 | 0.008 | √ | |
| 10 | I need a map showing the location of the Penfold's winery in Australia. | 19.31 | 14.35 | 0.014 | √ | |

statistically significant at more than 0 ($t$-test $p$-value $= 0.013$), which indicates that the effect of using AS over direct use of search engine was stronger for the tough tasks. This difference was due to the poorer QBS performance for the tough tasks ($t$-test $p$-value $= 0.00043$). The AS performance was not significantly different between the two groups.

### 4.7. Why adaptive search was effective: examples with explanations

We observed several examples confirming our conjectures about the effectiveness of adaptive search:

(1) *The system was able to pull out additional good terms in order to describe the clusters detected in the search results*. For example, given the task "I'm looking for the names of campgrounds around Lake Louise (Canada) that have showers," the system used the terms `campgrounds`, `trailer parks`, `camping`, `parks`, `campground`, and `facilities` for the summary of the search results, eliciting user opinion about the relevance of those terms to the information need.

(2) *Users were able to quickly distinguish terms explaining relevant or irrelevant documents, and correctly mark those terms*. For example, the users correctly marked the terms mentioned in the preceding paragraph as "close to" the information need. Users also correctly marked as "far from" the information need the word `sports` for the Star Ferry question.

(3) *The system is able to deliver to the user an efficient summary of current search results, permitting an easy check for omission relevant to the information need. The user was able to describe the missing information*. For example, while working on the task "Where can I get a good pfeffersteak in Hagerstown, MD USA?" and being presented by the system with the summary shown on Fig. 9, most users entered the word `pfeffersteak` on the additional input line since the summary did not mention it.

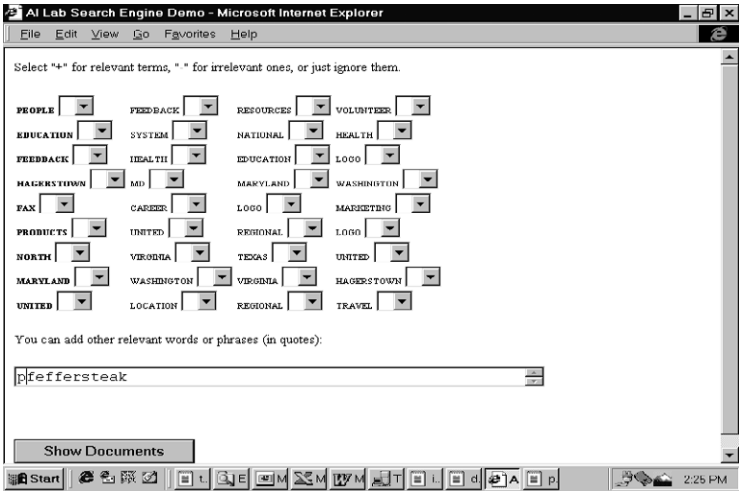


Fig. 9. The feedback form for the tast "Where can I get a good pfeffersteak in Hagerstown, MD USA?" Subject adding word "pfeffersteak" since it was not in the summary of clusters.

## 5. Conclusions, limitations and future research

Based on the statistical evidence described in the preceding section and our observations, our suggested approach to seeking information by clustering query results and summarizing seems promising, even superior to the traditional query based approach. Our approach has many additional benefits: it requires neither knowing a Boolean query language nor skill in selecting appropriate keywords. Subjects in the evaluation experiment found it intuitive, easy to use, and preferred using it to the traditional approach.

In our experiment, subjects searched the entire World Wide Web both by using our approach and using the search engine directly. Since our current implementation acts as a layer between the user and a commercial keyword-based search engine, we can conclude that subjects achieved better search effectiveness using AS because of the presence of this clustering and summarization layer. The experimental setup was such that the choice of a particular underlying search engine is not likely to be crucial to the outcome so long as the engine provides basic features, mentioned in Section 3.1.

As a result of the experimental design and our system architecture, we have been able to quantitatively test only the overall effect. We have not been able to examine the separate contributions of certain effects such as the following. First, since the set of clusters acts as a summary, users can recognize what is missing in it and supply the missing information. Second, the summary of clusters described by terms elicits the user's opinion on the relevance of those terms to the information need. Third, the interactive search tool encourages entering a textual description of the information need at the beginning of the search process, so the system acquires a better description of the context from the user. (AltaVista, the Internet search engine used by the control group, also allows, but does not require, starting with a textual description.)

Separate effects can be tested in follow-up studies. We believe that our empirical result that the *combination of these effects facilitates information access* is itself valuable.

Although resembling *pseudo-relevance feedback* (Rocchio, 1971), our approach essentially differs from it in that, rather than relying on user feedback, the system simply assumes that the top-ranked retrieved documents are relevant and uses them to augment the query with a relevance feedback ranking algorithm. Pseudo-relevance may augment the ranking algorithm that the underlying Web search engine uses. Our approach instead provides an interactive intermediate layer between the user and the search engine. Using our approach, pseudo-relevance feedback may be implemented independently inside the search engine and be completely hidden from the user.

The small number of tasks and their individuality are limitations of our study. However, as we mentioned in the literature review, we are not aware of large-scale studies of Web searching. We also believe that a strong point of our empirical set-up is that, contrary to previous studies, we did not compose queries but asked users to do it.

Imposing tasks on subjects has its limitations. However, we argue that this may represent a fairly typical situation such as when a supervisor asks a graduate assistant to find something related to a class, or a manager asks subordinates to find information crucial to a company's performance. Unfortunately, due to current implementation and Internet bandwidth, we had to pre-fetch Web pages to build the maps and thus were unable to use subjects' own tasks in the study.

Since our current implementation acts as a layer between the user and commercial keyword-based search engine, the output of user feedback is limited to constructing queries for the engine.

In future, we may be able integrate more tightly the summarizing layer and the retrieval layer to utilize the user feedback not only at the ''terms level'' but at the ''cluster level''. New algorithms may be developed to support this interaction.

Our data set consists of more than 36 h of recorded browsing behavior and thousands of visited web pages, which may be an excellent source for testing future hypotheses related to information seeking models. We intend to test the effects of graphical representation of clusters as opposed to HTML forms.

We believe that our study has several implications related to the dissemination of knowledge. One of our findings is that there exist on the web answers to even such specific questions as: ''Where can I get a good pfeffersteak in Hagerstown, MD USA?'' or ''How long does it take to get by train from Copenhagen to Oslo?'' We also found that, armed with the right tools, even novice users of searching technology can unearth those answers. We hope that, once convinced that they can find information.

## Acknowledgements

This research was supported by the following grants: NSF/DARPA/NASA Digital Library Initiative, IRI-9411318, and DARPA Information Management Program, N66001-97-C-8535.

## References

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

Bowman, C. M. (1994). The Harvest information discovery and access system. In *Proceedings of the second international world wide web conference'94*. Chicago, IL.

Buckley, C., Salton, G., & Allan, J. (1994). The effect of adding relevance information in a relevance feedback environment. In *Proceedings of the 17th annual international ACM SIGIR conference on research and development in information retrieval*, (pp. 292–300). Dublin, Ireland.

Card, S. K., Robertson, G. G., & York, W. (1996). The WebBook and the Web Forager: An information workspace for the world wide web. In *Proceedings of the ACM/SIGCHI conference on human factors in computing systems* (pp. 111–119). Vancouver, Canada.

Chen, H. (1994). Machine learning for information retrieval: Neural networks symbolic learning and genetic algorithms. *Journal of the American Society for Information Science*, *46*(3), 194–216.

Chen, H., Schuffels, C., & Orwig, R. (1996). Internet categorization and search: A self-organizing approach. *Journal of Visual Communication and Image Representation*, *7*(1), 88–102.

Chu, H., & Rosenthal, M. (1996). Search engines for the world wide web: A comparative study and evaluation on methodology. In *ASIA 1996 annual conference proceedings* (pp. 127–125). Baltimore, MD.

Cohen, A. C. (1959). Simplified estimators for the normal distribution when samples are singly censored or truncated. *Technometrics*, *1*(3), 217–237.

Cooper, J. W., & Byrd, R. J. (1997). Lexical navigation: Visually prompted query expansion and refinement. In *Proceeding of the second ACM international conference on digital libraries*, July 1997, (pp. 237–246). Philadelphia, PA.

Croft, W.B., Das, R. (1990). Experiments with query acquisition and use in document retrieval systems. In *Proceedings of the 13th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 349–368).

Cutting, D.R., Karger, D.R., Pedersen, J.O., & Tukey, J.W. (1992). Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of the 15th annual international ACM conference on research and development in information retrieval* (pp. 318–329).

DeBra, P. & Post, R. (1994). Information retrieval in the world wide web: Making client-based searching feasible. In *Proceedings of the first international world wide web conference'94*. Geneva, Switzerland.

Ding, W. & Marchionini, G. (1996). A comparative study of web search service performance. In *ASIS 1996 annual conference proceedings* (pp. 136–142). Baltimore, MD.

Gauch, S. & Guijin, W. (1996). Information fusion with profusion. In *Webnet'96 conference*. San Francisco, CA.

Harman, D. (1988). Towards interactive query expansion. In *Proceedings of 1lth annual international ACM SIGIR conference on research and development in information retrieval* (pp. 321–331). Grenoble, France.

Hearst, M. A. (1999). User interfaces and visualization. In R. Baeza-Yates, B. Ribeiro-Neto (Eds.) *Modern information retrieval*. New York: Addison-Wesley.

Hearst, M.A., & Pedersen, J.O. (1996). Reexamining the cluster hypothesis: Scatter/Gather on retrieval results. In *Proceedings of the 19th annual international ACM conference on research and development in information retrieval* (pp. 76–84). Zurich, Switzerland.

Jain, A. K., & Dubes, R. C. (1988). *Algorithms for clustering data*. New York: Prentice-Hall.

Jansen, B., Spink, A., Bateman, J., & Saracevic, T. (1998). Real life information retrieval. *SIGIR Forum*, *32*(1), 5–17 A publication of special interest group on information retrieval.

Koenemann, J., & Belkin, N. (1996). A case for interaction: A study of interactive information retrieval behavior and effectiveness. In *Proceedings of ACM conference on human factors in computing systems*, Vol. 1 (pp. 205–212). Zurich, Switzerland.

Kohonen, T. (1995). *Self-organizing maps*. Berlin: Springer.

Lagergren, E., & Over, P. (1998). Comparing interactive information retrieval systems across sites: The trec-6 interactive track matrix experiment. In *Proceedings of the 21st annual international ACM/SIGIR conference* (pp. 164–172).

Lawrence, S., & Giles, C. L. (1999). Accessibility of information on the web. *Nature*, *400*, 107–109.

Magennis, M., & van Rijsbergen, C. (1997). The potential and actual effectiveness of interactive query expansion ACM SIGIR Forum. In *Proceedings of the 20th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 324–332). Philadelphia, PA.

Mechkour, M., Harper, D., & Muresan, G. (1998). The WebCluster project. using clustering for mediating access to the world wide web. In *Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval* (pp. 357–358). Melbourne Australia.

Orwig, R. E., Chen, H., & Nunamaker, J. F. (1997). A graphical self-organizing approach to classifying electronic meeting output. *Journal of the American Society for Information Science*, *48*(2), 157–170.

Pinkerton, B. (1994). Finding what people want: Experiences with the WebCrawler. *Proceedings of the second international world wide web conference'94*. Chicago, IL.

Pirolli, P., Schank, P., Hearst, M.A. & Diehl, C. (1996). Scatter/Gather browsing communicates the topics structure of a very large text collection. In *Proceedings of the ACM CHI96 conference*.

Rocchio, J. J. (1971). Relevance feedback in information retrieval. In G. Salton (Ed.), *The SMART retrieval system* (pp. 313–323). Englewood Cliffs, NJ: Prentice-Hall.

Roussinov, D., & Chen, H. (1998). A scalable self-organizing map algorithm for textual classification: A neural network approach to thesaurus generation. *Communication and Cognition – Artificial Intelligence*, *15*(1-2), 81–112.

Roussinov, D., Tolle, K., Ramsey, M., & Chen, H. (1999). Interactive internet search through automatic clustering: An empirical study. In *Proceedings of 22nd ACM SIGIR*, 15–19 August 1999. Berkeley, CA (poster).

Roussinov, D., & Ramsey, M. (1998). Information forage through adaptive visualization. In *Proceedings of the third ACM conference on digital libraries* (pp. 303–304), 23–24 June 1998. Pittsburgh, PA.

Sahami, M., Yusufali, S., & Baldonado, Q.W. (1998). SONIA: A service for organizing networked information autonomously. In *Proceeding of the third ACM international conference on digital libraries* (pp. 237–246). Pittsburgh.

Salton, G., & Buckley, C. (1990). Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, *41*(4), 288–297.

Salton, G., & McGill, M. J. (1983). *Introduction to modern information retrieval*. New York: McGraw-Hill.

Smeaton, A. F., & van Rijsbergen, C. J. (1983). The effects of query expansion on a feedback document retrieval system. *Computer Journal*, *26*, 239–246.

Spink, A. (1997). Study of interactive feedback during mediated information retrieval. *Journal of the American Society of Information Science*, *48*(5), 382–394.

Voorhees, E.M. & Harman, D.K. (1999). The eighth text retrieval conference (TREC-8). In *Proceedings of the eighth text retrieval conference (TREC-8)*. Gaithersburg, Maryland, 17–19 November 1999.

Wang Baldonado, M.Q., & Winograd, T. (1997). SenseMaker: An information-exploration interface supporting the contextual evolution of a user's interests. In *Proceedings of the ACM/SIGCHI conference on human factors in computing systems* (pp. 11–18). Atlanta, GA.

Westera, G. (1996). Search engine comparison: Testing retrieval and accuracy. [online] http://www.curtin.edu.au/curtin/library/staffpages/gwpersonal/senginestudy/results.htm.

Wiener, N. (1948). *Cybernetics: On control and communication in the animal and the machine*. Cambridge, MA: MIT.

Young, D., & Shneiderman, B. (1993). A graphical filter/flow model for Boolean queries An implementation and experiment. *Journal of the American Society for Information Science*, *44*, 327–339.

Zamir, O., Etzioni, O., Madani, O., & Karp, R.M. (1997). Fast and intuitive clustering of Web documents. In *Proceedings of the third international conference on knowledge discovery and data mining* (pp. 287–290).