2000

Information Navigation by Clustering and Summarizing Query Results

Dmitri G. Roussinov
Syracuse University

Michael J. McQuaid
University of Arizona

Recommended Citation

Roussinov, D., and McQuaid, M.J., Information Navigation on the Web by Clustering and Summarizing Query Results, Information Processing and Management (IP&M8), 37 (6), 2001, pp. 789-816.

Information Navigation by Clustering and Summarizing Query Results

Dmitri G. Roussinov
Syracuse University, School of Information Studies
droussin@syr.edu

Michael J. McQuaid
University of Arizona, Department of Management Information Systems
mcquaid@bpa.arizona.edu

Abstract

We have explored and evaluated a novel approach to information seeking grounded in the idea of summarizing query results through automated document clustering. The user starts with a natural language description of the needed information and navigates the information space through the interaction with the system.

We implemented a prototype allowing searches of a significant portion of the entire World Wide Web. In a laboratory experiment, subjects searched the WWW for answers to a given set of questions. Our results indicate that our prototype improved search performance, presumably through better understanding of query results. In addition, we analyzed interaction patterns and the effects of such parameters as subject skills and task peculiarities.

1. Motivation

1.1. Query based search results in information overload

As firms rely increasingly on Intranets [26], enterprise information resources proliferate in forms already familiar on the WWW [2] along with information overload problems familiar on the Web. Knowledge workers gain the ability to search corporate resources in the same manner and with the same limitations as they search Internet resources. Opportunities for knowledge management in the Intranet context have been proposed [1] [8] [17]. These opportunities depend on technologies for search and visualization [18]. Although the bulk of organizational data is currently in text format, the ability to search it remains inadequate [9].

In the realm of the traditional keyword approach to information seeking, we need to match words in the sought documents 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?", seems 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. Most users tolerate exploring only the first 20-30 pages in the ranked lists returned by search engines and give up.

People have great difficulty specifying queries in Boolean format and often misjudge what the results will be [11] [28]. Boolean queries may be problematic for several reasons. Most people find the basic syntax counter-intuitive. Many users confuse everyday semantics associated with Boolean operators. For example, to inexperienced users, using AND implies the widening of the scope of the query: “stocks AND bonds” may be interpreted as a request for documents about “stocks” and documents about “bonds,” rather than documents that talk about both “stocks” and “bonds.”

1.2. Interactive approach to information seeking

People read through textual information much more slowly than computers, which can sift through megabytes of text in a moment. Unfortunately for their masters, computers do not have the human notion of common sense. They do not understand natural language variations, and so are unlikely to autonomously find answers to many questions posed by humans. Besides, people are often vague and need several reiterations to describe their information needs even to other people. When users approach an information access system they often have only a fuzzy understanding of how to achieve their goals. It is not surprising that iterative clarifying is required while interacting with a computerized information access system [23].

Clustering and summarizing query results have been proposed as potentially effective tools facilitating interactive information access [11]. The idea behind them is to agglomerate similar documents into clusters and present a high-level summary of each cluster. This way, the user does not need to peruse similar documents nor the full length of documents to become familiar with the subset of documents. Examples of systems based on query result summarization and reformulation are Scatter/Gather [12], Lexical Navigation [7], WebBook [3], and SenseMaker [27]. Hearst [11] has noted that no empirical evidence has been tendered that any of these tools facilitates information access while negative results have proliferated.

1.3. Research question

In this study, we first speculated on why prior experience showed negative results with interactive information access systems using document

0-7695-0493-0/00 $10.00 (c) 2000 IEEE

clustering/summarization. We detail this in the next subsection. Based on our conjectures and prior studies, we designed an approach called *Adaptive Search*, which is based on a novel use of clustering, summarization and user feedback. We empirically tested *whether our approach is superior to the traditional query based approach*. Since our approach falls into a broad class of interactive information access based on document clustering, and no evidence of the effectiveness of that class has been tendered so far, we also concomitantly address a higher level question: *does automated document clustering facilitate interactive information access?*

Since our approach is implemented on top of a query based search engine (AltaVista in this particular study), to test if our layer improves search performance, we ran a controlled experiment and compared using our approach against using the search engine directly. We also observed and explored user behavior while experiencing both approaches. We analyzed our results to find out for what kinds of tasks and what kinds of users each approach is more effective.

1.4. Improving the approach

We have followed a previously suggested idea of summarizing current query results [12] [22]. We do so by automated clustering. We conjecture that *clustering only the query results* has an advantage over clustering the entire collection since it is specific to the task at hand. In a way, it *adapts* to a particular user (novice or expert) and a particular search. According to cognitive fit theory [25], the mental representation of a task and the presentation of information relevant to the task must fit or the user is forced to sacrifice time or accuracy to compensate. We should be able to see time or accuracy degrade for information presentations offering a worse fit.

Studies have shown [6] that users easily become disoriented while navigating hierarchies, both manually created such as Yahoo as well as automatically generated such as self-organizing maps. Navigating requires remembering which path is being currently pursued. It may be time consuming and frustrating if the desired information is deep in the hierarchy or not available at all. A wrong path requires backing up and trying again. On the result side, presenting documents in rank order has become standard for modern search systems including Internet spiders. 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*.

Since prior studies have shown [21] that clusters do convey a high level picture of document collections, we conjecture that the user may be able to provide feedback to the system by looking at the clusters of query results without actually going through time-consuming exploration of documents. 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 in this summary*. Many interactive systems based on relevance feedback support types 1 and 2 [11]. However, we believe that supporting type 3 is also vital for success. It justifies the application of clustering techniques to create a summary. We conjectured that simply displaying the most frequent terms in the query result might not adequately highlight missing items. The next section explains how we implemented our feedback mechanism.

Our study used Kohonen's self-organizing maps (SOM) [14] as a clustering and summarization tool. Developed in the 80s, 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 et al. [6] showed that SOM could efficiently summarize a collection of text documents. Orwig et al. [19] showed that SOM could identify key concepts mentioned in a collection of text documents. Being an unsupervised neural network, SOM is known to be more robust to noisy inputs [4], such as text documents usually are, than statistical clustering techniques such as single-link or K-means [13].

Adaptive Search does not require familiarity with a query language. Our approach differs in this way from query formulation interfaces [20], [16]. We see our approach as a first step towards a more general *information navigation* paradigm, in which the user only directs the system toward the documents satisfying the information need, and does not need to think in terms of boolean queries. The user starts with a simple natural language description of the information need, which has been proposed as beneficial for the bulk of consumers of modern information services [10].

2. Adaptive search implementation

2.1. Commercial Search Engines Capabilities

Most of the 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 the 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*. Users of AltaVista “Simple Search” do this by simply entering those words or phrases in the query. For example, the query “hong kong tsimshatsui” results 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 leads to a higher chance for the page to be at the beginning. Search engines also use metadata to influence the rank order. For example, the presence of those words in the title would rank the documents 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 all of the words "hong" and "kong". This is equivalent to the "AND" operator in many Boolean query languages: "hong AND kong." In the "Simple Search," words preceded by "+" sign also affect rank order as explained in the preceding paragraph. So, not only the engine returns pages containing all of those words, but also it orders them such that pages containing many of those words would appear at the beginning.

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 to refine their queries. This number being very large indicates that the word is too general and probably not very useful in queries. This number being very small or zero usually indicates spelling errors or other reasons for the word's unpopularity on the web.

2.2. Interaction between the user and the system

Our system is implemented as an additional layer between the user and any commercial search engine (AltaVista). The interaction between the user, the system and the search engine is shown in Figure 1. It proceeds through the following steps:

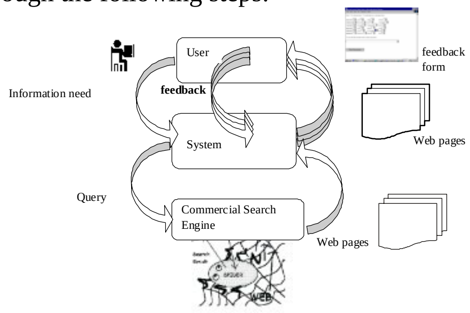


Figure 1. Adaptive Search approach.

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 list in ranked order 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 [14]. The self-organizing map algorithm automatically clusters documents and assigns labels to the clusters. Based on the information contained in the map, the system generates a summary presented as an HTML form to the user. 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 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 their snippets, which the commercial search engine provides.

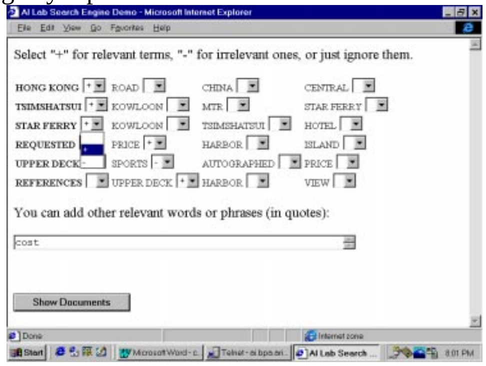
Step 6. Iterate: the user may fill out the feedback form differently and re-submit it, thus going through steps 2-5 repeatedly in order to find the pages of interest.

2.3. Summary and feedback form

The summary/feedback form shown in Figure 2 is a simplified textual representation of a Kohonen's self-organizing map (SOM). SOM accepts objects described by their features as inputs and places those objects on a 2D grid in such a way that similar objects are places nearby. In this sense it is similar to multidimensional scaling [13]. More details about application of SOM to summarizing text documents can be found in [19]. SOM also creates clusters of documents (called *regions*), each labeled with a unique word or phrase.

Figure 2 shows an example of an HTML form used in Adaptive Search. 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 "+" or "-" sign from a pull-down menu. The descriptive term list (6 × 4 = 24 in the example in figure 2) doublefunctions as a document summary: the user can add any missing key topics on the bottom line.



Select "+" for relevant terms, "-" for irrelevant ones, or just ignore them.

HONG KONG	+	ROAD		CHINA		CENTRAL	
TSIMSHATSUI	+	KOWLOON		MTR		STAR FERRY	
STAR FERRY	+	KOWLOON		TSIMSHATSUI		HOTEL	
REQUESTED		PRICE	+	HARBOR		ISLAND	
UPPER DECK		SPORTS	-	AUTOGRAPHED		PRICE	
REFERENCES		UPPER DECK	+	HARBOR		VIEW	

You can add other relevant words or phrases (in quotes):

cost

Show Documents

Figure 2. An example of an HTML form generated by Adaptive Search for the Star Ferry task

2.4. The user feedback component

After a number of preliminary studies, we eventually settled on a heuristic algorithm that gave the best preliminary results. The general idea behind the algorithm is to first 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 enough documents.

The algorithm constructed the "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". To achieve this, the algorithm used Feature 3 described 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). To achieve this, the algorithm used Feature 1 described 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 Figure 2. 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:

```
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 requires matching documents to have the words hong kong, tsimshatsui, upper deck, price, and star ferry. The matching documents will 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 modifies the "ideal query" by removing the required (prefixed with "+" in the ideal query) or excluded (prefixed with "-") words/phrases that are also very frequent from the query. The frequencies of occurrence of a word or a phrase throughout the web (called *document frequencies*) are 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.

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

The table indicates that the phrase star ferry is very rare, while the word price is quite common. So the algorithm would drop the term price before star ferry. If several terms had to be dropped, the algorithm followed a heuristic strategy based on the widely used "inverse document frequency" weighting [24]. The strategy of our algorithm was to maximize the objective function, which was the sum of $\log (N / df_i)$ for each term i remaining in the query. Here, df_i is the number of web pages containing the term i (document frequency) and N is the maximum df_i among all the terms in the "ideal" query: $N = \max \{ df_i \}$, for i = 1 to M, where M is the number of required words and phrases in the "ideal" query. This ensures that the logarithm above always exists. This strategy would discard the most frequently occurring words, since they would have smaller weights.

3. Empirical evaluation

This section describes an empirical study comparing Adaptive Search (AS) approach and direct use of Internet search engine, which we denote as a Query Based Search

Table 2. Overall statistical results: Adaptive Search vs. Query Based Search

metric average	Query Based Search	Adaptive Search	hypothesis	t-test, p-value
user time	8.7 minutes	7.3 minutes	reject H1	0.08
physical time	13.1 minutes	11.9 minutes	can not reject H2	0.19
number of pages visited	5.07	4.17	reject H3	0.090
answer rank	12.86	10.60	reject H4	0.002
proportion of tasks accomplished	15/36	21/36		

(QBS). Thirty-six (36) undergraduate students in the school of business agreed to be subjects in this study. Their skill and experience with commercial search engines was pre-tested by the questionnaire.

3.1. 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. Since each search task was time consuming, each subject performed only 2 assigned tasks out of a total set of 10 search tasks used in the study. The tasks were randomly assigned. Each task was performed the same number of times with each tool. Order of tasks and interfaces was balanced to counter carry-over learning effects.

The subjects performed search tasks proposed by the panel on Web Search at the 1998 ACM Conference on Advances in Information Retrieval (listed below in Table 3). The search tasks were very specific and clearly defined, mostly related to entertainment/tourism topics. For example: “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” situations in which only 2-10 pages on the web contain the answer to the questions. Since the tasks were very unambiguous it was easy for the supervisor to decide whether the answer had been found.

3.2. Procedure

Subjects performed two kinds of search tasks, “real” and “virtual.” The “real” search tasks required the subject to find a web document containing an answer to the search question. The eight (8) “virtual” search tasks (the remaining tasks out of 10 tasks used in the study), required subjects to submit their queries (in the case of QBS) or the feedback form (in case of AS) once only. Virtual tasks were less time-consuming and thus more efficient in providing data. The “Metrics” section explains how these data have been used.

The experimental procedure with each subject was the following:

1. Fill out pre-questionnaire. (5 minutes)
2. Perform tutorial. (10 minutes)
3. Perform 1st task with interface 1 (QBS or AS). (up to 20 minutes)
4. Perform 2nd task with interface 2 (AS or QBS). (up to 20 minutes)
5. Perform 8 virtual tasks while switching interfaces. (approximately 10 minutes)
6. Fill out post-questionnaire. (5 minutes)

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 do repeated form submission (step 6).

If the subject used Query Based Search, the subject was free to enter a query using AltaVista “simple search” syntax, which allows entering the text description of the task unaltered as well. Entering entire text description never resulted in finding the answers. 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.

3.3. Metrics

We measured the time it took to find the answer to a given question. We assumed that a better system takes less time for information seeking.

We analyzed our log files after pilot experiments and came to the conclusion that the Adaptive Search system itself spent considerably more time processing, 95% of it for multiple interaction sessions with the commercial search engine (AltaVista). This time would not be necessary if Adaptive Search were located on the same server as the search engine. We believe that more accurate metrics should derive from the time that the user spends

Table 3. Average answer rank for each search task.

Task	Task Description	Average Answer Rank		t-test, p-value	Adaptive Search 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	√	

searching, ignoring waiting time, for two reasons. First, waiting time is a function of underlying technology and may be easily reduced. Second, the cognitive load during waiting is not as high as during active interaction.

This explains our choice of user search time, obtained from our log files, as our primary metric. We limited each session to 13 minutes (user time). Although we allowed some subjects to spend more time searching, we truncated to 13 minutes any time that was greater than 13 minutes for analysis purposes and considered the task accomplished within the time allowed only if it was accomplished within 13 minutes (user time). We chose the threshold to be 13 because that was the minimum user time of all unaccomplished tasks.

To evaluate the quality of the returned rank ordered lists of documents, we used a different 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 the 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* [24] because of our main purpose. We were interested in having the subject find at least one document containing the answer to the search task, but not 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 [15].

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.

Since we limited the time subjects could spend searching, we believed another useful metric to be the proportion of tasks accomplished in the given time.

3.4. 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 takes the same number of pages to visit in order to find the answer.

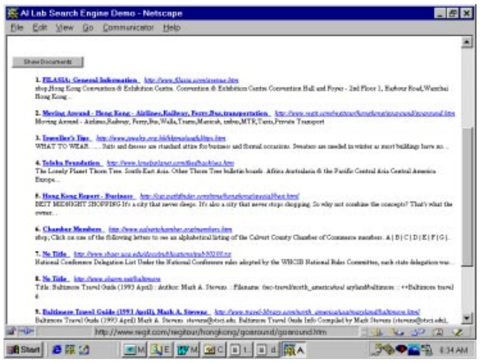
H4: Using both tools results in the same AnswerRank.

The alternative hypotheses were that AS was better as revealed by the metrics described above.

3.5. Example

This section presents an example of a 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 Figure 2. As it turns 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 Figure 3. The second web page contains the passage shown in Figure 4, which clearly contains the answer to the search question. Another subject was asked to use a query based search form and entered:

```
"hong kong" + "Star Ferry"
```


AI Lab Search Engine Demo - Netscape
File Edit View Go Communicator Help
Show Documents
Start

1. FILASIA: General Information [illegible]
sbop,Hong Kong Convention & Exhibition Centre. Convention & Exhibition Centre Convention Hall and Foyer - 2nd Floor 1, Harbour Road,Wanchai Hong Kong...

2. Moving Around - Hong Kong - Airlines,Railway, Ferry,Bus,transportation [illegible]
Moving Around - Airlines,Railway, Ferry,Bus,Walla,Trams,Maxicab, subway,MTR,Taxis,Private Transport

3. Traveller's Tips [illegible]
WHAT TO WEAR...... Suits and dresses are standard attire for business and formal occasions. Sweaters are needed in winter as most buildings have no...

4. Teleha Foundation [illegible]
The Lonely Planet Thorn Tree. South-East Asia. Other Thorn Tree bulletin boards. Africa Australasia & the Pacific Central Asia Central America Europe...

5. Hong Kong Report - Business [illegible]
BEST MIDNIGHT SHOPPING It's a city that never sleeps. It's also a city that never stops shopping. So why not combine the concepts? That's what the owner...

6. Chamber Members http://www.calvertchamber.org/members.htm
sbop; Click on one of the following letters to see an alphabetical listing of the Calvert County Chamber of Commerce members. A | B | C | D | E | F | G |...

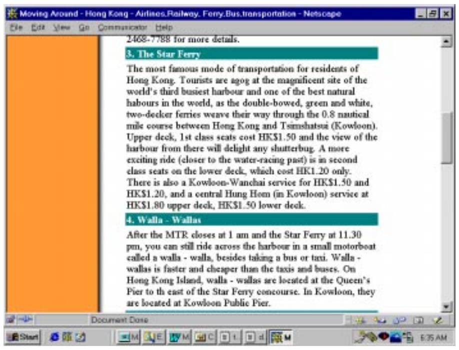
7. No Title [illegible]
National Conference Delegation List Under the National Conference rules adopted by the WHCIB National Rules Committee, each state delegation was...

8. No Title http://www.charm.net/baltimore
Title: Baltimore Travel Guide (1993 April) : Author: Mark A. Stevens : : Filename: (rec-travel/north_america/usa/maryland/baltimore : : ++Baltimore travel g

9. Baltimore Travel Guide (1993 April), Mark A. Stevens [illegible]
Baltimore Travel Guide (1993 April) Mark A. Stevens. stevens@bsci.edu. Baltimore Travel Guide Info Compiled by Mark Stevens (stevens@bsci.edu).

http://www.regit.com/regitour/hongkong/goaround/goaround.htm

Figure 3. A list of documents found by Adaptive Search for the Star Ferry question.


Moving Around - Hong Kong - Airlines,Railway, Ferry,Bus,transportation - Netscape
File Edit View Go Communicator Help
Document Done
Start
6:35 AM

2468-7788 for more details.

3. The Star Ferry

The most famous mode of transportation for residents of Hong Kong. Tourists are agog at the magnificent site of the world's third busiest harbour and one of the best natural habours in the world, as the double-bowed, green and white, two-decker ferries weave their way through the 0.8 nautical mile course between Hong Kong and Tsimshatsui (Kowloon). Upper deck, 1st class seats cost HK$1.50 and the view of the harbour from there will delight any shutterbug. A more exciting ride (closer to the water-racing past) is in second class seats on the lower deck, which cost HK1.20 only. There is also a Kowloon-Wanchai service for HK$1.50 and HK$1.20, and a central Hung Hom (in Kowloon) service at HK$1.80 upper deck, HK$1.50 lower deck.

4. Walla - Wallas

After the MTR closes at 1 am and the Star Ferry at 11.30 pm, you can still ride across the harbour in a small motorboat called a walla - walla, besides taking a bus or taxi. Walla - wallas is faster and cheaper than the taxis and buses. On Hong Kong Island, walla - wallas are located at the Queen's Pier to th east of the Star Ferry concourse. In Kowloon, they are located at Kowloon Public Pier.

Figure 4. A passage from a web page containing the answer to the Star Ferry question.

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 the ticket on the upper deck or Tsimshatsui as 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 [28]: forgetting keywords, mismatched vocabularies, wrong use of syntax, too short or too long queries that result in too many or no documents.

3.6. Results and discussion

Our data are available on the web (http://ai.bpa.arizona.edu/resume/dmitri/current.dat). Table 2 display several overall statistical results that are discussed below in this section. We have to reject null hypothesis 1. This result provides strong evidence that Adaptive Search requires less time, despite high involvement of random factors such as differences in web transmission times, subject skills, and task difficulties.

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, different amount of typing while working with different interfaces is extremely unlikely to explain the above result.

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 Answer Ranks, we have analyzed about 40*10*20 = 8000 pages, including some duplicates. This analysis has taken approximately 80 person-hours total. The person responsible for analyzing pages did not know which system each particular page came from, so there was no bias towards one of the systems. We rejected the hypothesis 4 and concluded that Adaptive Search consistently positioned the user closer to finding the answer than Query Based search did.

We observed that many subjects wrongfully 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 contribute to diminishing the time difference even when Adaptive Search consistently provides better lists of matching documents.

Most of the subjects indicated in a questionnaire that they preferred Adaptive Search to Query-Based search. We asked them to choose an integer number between 1 and 5 to describe their preference between two tools, 1 corresponding to the strongest preference of QBS and 5 to the strongest preference of AS. The average subject preference has turned out to be 3.6, which is statistically significantly more than 3 (corresponding to indifference between those two tools) with t-test p-value = 0.003. Some reasons subjects gave for their preference for Adaptive Search over Query Based search include the following. 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 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 Answer Rank. 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 1/2 and 1/3, so the exact cut-off number is much less important. We obtained the same result for the Reciprocal Answer Rank (RAR), with t-test p-value = 0.011.

To address the concern of 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:

effect = average RAR using AS - average RAR using QBS.

The effects are independent. They are displayed in the Figure 5. The mean effect (0.12) was positive (t-test p-value = 0.013).

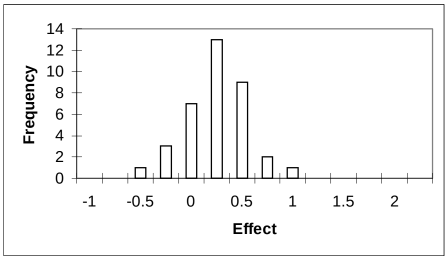


Figure 5. Histogram of the effect by subjects.

Notably, the average answer rank 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 "real" sessions considered), t-test p-value = 0.0026. Thus, even after looking at the rank ordered list and sometimes 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 those using AS were by a single submission of the HTML form.

In order to find out for what types of users the effect would be stronger, we divided subjects into two groups of equal size by their average performance with both tools for the analysis purposes. We found that the effect is stronger for the lower half: the means are significantly different with t-test p-value = 0.016. This suggests that 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 in which the subject was recruited (from introductory class to a 3rd year core class). Again, we distinguished two groups of subjects (sizes 17 and 14; 5 ignored 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" group higher (0.35 vs. 0.27) as measured by average reciprocal answer rank (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. To reconcile, we suggest that the class may not be a reliable predictor of the overall performance and the effect.

Table 3 shows average Answer Rank for all 10 tasks. Answer Rank was significantly better for Adaptive Search in the tasks 2, 3, 9, 10; worse in the task 4 and not statistically different in tasks 1, 5, 7. Answer Rank was the same and invariably equaled 20 for the tasks 6 and 8, since no subjects were able to obtain answers in 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 the task selection, as long as tasks remain at the same level of difficulty and peculiarity ("needle in a haystack").

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 found answer-pages through pretests and the experiment, and ignored tasks 6 and 8 since no subjects found answers to them.

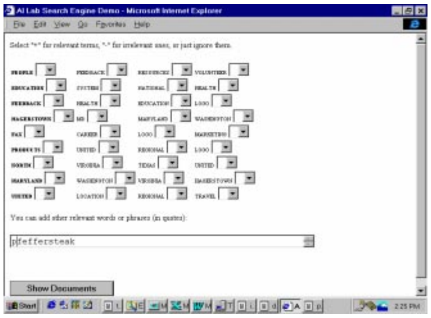
For each subject, we computed the difference in effects between those measured by the "tough" tasks (1,3,9,10) and by the "easy" tasks (2,4,5,7). The mean difference was found to be statistically significantly more than 0 (t-test p-value = 0.013), which indicates that the effect of using AS over direct use of search engine is stronger for the "tough" tasks. This difference was due to the worse QBS performance for the "tough" tasks (t-test p-value = 0.00043). The AS performance was not significantly different between those two groups.

3.7. Why adaptive search was effective: qualitative analysis

We have witnessed 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 are 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 were also correctly marking words as "far from" information need, as in the case of the word `sports` for the Star Ferry question.


AI Lab Search Engine Demo - Microsoft Internet Explorer
File Edit View Go Favorites Help
Show Documents
Start
2:25 PM

Select "+" for relevant terms, "-" for irrelevant ones, or just ignore them.

PEOPLE	FEEDBACK	RESOURCES	VOLUNTEER
EDUCATION	SYSTEMS	NATIONAL	HEALTH
FEEDBACK	HEALTH	EDUCATION	LOGO
HAGERSTOWN	MD	MARYLAND	WASHINGTON
FAX	CAREER	LOGO	MARKETING
PRODUCTS	UNITED	REGIONAL	LOGO
NORTH	VIRGINIA	TEXAS	UNITED
MARYLAND	WASHINGTON	VIRGINIA	HAGERSTOWN
VISITED	LOCATION	REGIONAL	TRAVEL

You can add other relevant words or phrases (in quotes):

pfeffersteak

Figure 6. The feedback form for the task "Where can I get a good pfeffersteak in Hagerstown, MD USA?" Subject adding word "pfeffersteak" since it was not in the summary of clusters.

3. The system is able to deliver an efficient summary of current search results to the user, permitting an easy check for omission relevant to the information need. The user is able to describe the missing information. For example, while working in the task "Where can I get a good pfeffersteak in Hagerstown, MD USA?" and being presented by the system with a summary shown on Figure 6, most users entered the word `pfeffersteak` on the additional input line since the summary did not mention pfeffersteak.

4. Conclusions, limitations and future research

Based on the statistical evidence described in the preceding section and our observations, our suggested approach to information seeking based on query results clustering 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 using our approach (treatment group) and the search engine directly (control group). Since our current implementation acts as a layer between the user and a commercial keyword based search engine, we can conclude that the better search effectiveness revealed by subjects using Adaptive Search is due to this layer, which is based on clustering and summarization. Due to this experimental setup, the choice of particular underlying search engine is not likely to be crucial for the outcome as long as the engine provides basic features, mentioned in the section 2.1.

Due to the design of our experiment and our system architecture, we have been able to quantitatively test only the overall effect. We have not distinguished between the separate contributions of several effects as follows. First, the set of clusters acts as a summary, so users recognize what is missing in this summary and describe 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 more context description from the user. (AltaVista, the Internet search engine used on control group, also allows, but does not require, starting with textual description.) Fourth, interactive search tools encourage more user input, leading to better results.

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.

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 hours 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.

5. Acknowledgements

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

6. References

[1] Ba, S.L., Lang, K.R. and Whinston, A.B. (1997). Enterprise decision support using intranet technology. *Decision Support Systems* 20(2) 99-134, Jun 1997.

[2] Bannan, J. (1997) Intranet document management. Addison-Wesley Developers Press, Reading, Mass.

[3] Card, S.K., Robertson, G.G., & York, W. (1996). The WebBook and the Web Forager: An Information Workspace for the World-Wide Web. *Proceedings of the ACM/SIGCHI Conference on Human Factors in Computing Systems* (pp. 111-119). Vancouver.

[4] Chen, H. (1994). Machine Learning for Information Retrieval: Neural Networks, Symbolic Learning, and Genetic Algorithms. *Journal of the American Society for Information Science*.

[5] Chen, H. and Lynch, K.J. (1992). Automatic Construction of Networks of Concepts Characterizing Document Databases. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(5) (pp. 885-902).

[6] Chen, H., Schuffels, C., & Orwig, R. (1996). Internet Categorization and Search: A Self-Organizing Approach. *Journal of Visual Communication and Image Representation*, 7(1) (pp. 88-102).

[7] Cooper, J.W., & Byrd, R.J. (1997). Lexical Navigation: Visually Prompted Query Expansion and Refinement. Proceeding of the 2nd *ACM International Conference on Digital Libraries*, July 1997, (pp. 237-246).

[8] Davenport, T.H. and Prusak, L. (1998). Working Knowledge: How Organizations Manage What They Know. Harvard Business School Press, Boston, MA.

[9] Gordon. M. (1997). It's 10 a.m. Do You Know Where Your Documents Are? The Nature and Scope of Information Retrieval Problems in Business. *Information Processing and Management*, 33(1) (pp. 107-121).

[10] Gore, A., (1999). Information technology for the twenty-first century, *Year 2000 White House Budget Proposal*.

[11] Hearst, M.A. (1999). User Interfaces and Visualization**,** in *Modern Information Retrieval*, edited by Ricardo Baeza-Yates and Berthier Ribeiro-Neto, Addison-Wesley Publishing Company.

[12] Hearst, M.A., & Pedersen, J.O. (1996). Reexamining the Cluster Hypothesis: Scatter / Gather on Retrieval Results. *Proceedings of the Nineteenth Annual International ACM Conference on Research and Development in Information Retrieval* (pp. 76-84). Zurich.

[13] Jain, A.K., Dubes, R.C. *Algorithms for Clustering Data*. 1988. Prentice Hall.

[14] Kohonen, T. (1995). *Self-Organizing Maps*. Berlin, Heidelberg. Springer-Verlag.

[15] 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).

[16] Landauer, T.K., Egan, D.E., Remde, J.R., Lesk, M., Lochbaum, C.C., & Ketchum, D. (1993). Enhancing the usability of text through computer delivery and formative evaluation: the superbook project. *In Hypertext: A psychological perspective*, ed. by C. McKnight, A. Dillon, & J. Richardson (pp. 71-136). Ellis Horwood.

[17] Nonaka, I. (1994). A dynamic theory of organizational knowledge creation. *Organization Science*, 5(1) (pp. 14-37).

[18] O'Leary, D.E. (1998). Enterprise knowledge management. *IEEE Computer*, 31(3) (pp. 54-61).

[19] 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) (pp. 157-170).

[20] Pedersen, J.O., Schmeltz, G. (1993). A browser for bibliographic information retrieval, based on an application of lattice theory. In *Proceedings of the 16th Annual International ACM/SIGIR Conference*, (pp. 270-279), Pittsburgh, PA.

[21] Pirolli, P., Schank, P., Hearst, M.A.& Diehl, C. (1996). Scatter/Gather Browsing Communicates the Topics Structure of a Very Large Text Collection, *Proc. ACM CHI96 Conference*, April 13-18.

[22] Plaisant, C., Bruns, T., Shneiderman, B., & Doan, K. (1997). Query previews in networked information systems: the case of EOSDIS. In *Proceedings of ACM CHI 97 Conference on Human Factors in Computing Systems* (pp. 202-203).

[23] Rao, R., Pedersen, J.O., Hearst, M.A. & Mackinlay, J.D. (1995). *Rich Interaction in the Digital Library. Communications of the ACM*, 38(4) (pp. 29-39).

[24] Salton, G., & McGill, M.J. (1983). *Introduction to Modern Information Retrieval*. New York. McGraw-Hill.

[25] Vessey, I. (1991). Cognitive fit: Theory-based analyses of the graphs versus tables literature. *Decision Science*, 22(1), pp. 219-241.

[26] Wagner, R.L. and Engelmann, E. (1997). Building and managing the corporate intranet. McGraw-Hill, New York.

[27] Wang Baldonado, M.Q., & Winograd, T. (1997). SenseMaker: An Information-Exploration Interface Supporting the Contextual Evolution of a User's Interests. *Proceedings of the ACM/SIGCHI Conference on Human Factors in Computing Systems* (pp. 11-18). Atlanta, GA.

[28] 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 (pp. 327-339).