Looking Ahead: Query Preview in Exploratory Search

Pernilla Qvarfordt, Gene Golovchinsky, Tony Dunnigan
FX Palo Alto Laboratory, Inc.
3174 Porter Drive, Palo Alto, CA 94304
{pernilla, gene, tonyd}@fxpal.com

Elena Agapie
Harvard University
33 Oxford St, Cambridge, MA 02138
eagapie@seas.harvard.edu

ABSTRACT
Exploratory search is a complex, iterative information seeking activity that involves running multiple queries and finding and examining many documents. We designed a query preview control that visualizes the distribution of newly-retrieved and re-retrieved documents prior to running the query. When evaluating the preview control with a control condition, we found effects on both people's information seeking behavior and improved retrieval performance. People spent more time formulating a query and were more likely to explore search results more deeply, retrieved a more diverse set of documents, and found more different relevant documents when using the preview.

Categories and Subject Descriptors
H.5.m [**Information interfaces and presentation (e.g., HCI)]:** Miscellaneous.

Keywords
Information seeking, exploratory search, information retrieval, HCIR

1. INTRODUCTION
Exploratory search plays an important role in many domains such as academic research, intelligence analysis, e-discovery and pharmaceutical research. Information seeking in these fields typically involves long sessions consisting of many queries, evolving information needs as searchers learn about the topic of interest and about the collection, and a focus on finding many pertinent documents (not just one "best match").

Exploratory search is a complex, cognitively demanding activity that places a heavy load on memory and on sense-making processes. Forcing people to use external tools that are poorly integrated or requiring them to rely on memory for significant periods of time may make a difficult task even harder. On the other hand, an overly complex interface may impose its own cognitive burden, distracting from the real task. Thus one challenge in building tools to support exploratory search involves finding a sweet spot in the design space: making tools that help more than they distract.

In this paper, we describe a visualization that is designed to help people understand the relationship between the documents a query *will* retrieve and documents *already found* within in a search session. While searchers are formulating their query (e.g. typing in query terms or adding a document to the query as relevance feedback), a preview control displays the outcome of the query by aggregating the counts of new documents, of documents already retrieved, and of seen documents. This preview control helps people reformulate queries on the fly, without having to wait for the results to be incorporated into the workspace.

The contributions of this work are a description and implementation of a novel interface widget for facilitating exploratory search, and an experimental evaluation of the widget that assessed its impact on user behavior and system performance.

Below, after covering the related work, we describe the preview control, and report the results of a controlled study to assess its effectiveness. We conclude with a discussion of our findings and their implications for interface design for information seeking.

SIGIR'13, July 28–August 1, 2013, Dublin, Ireland.

2. RELATED WORK
Exploratory search is often recall-oriented, as searchers seek a more complete description of particular ideas or phenomena [15]. The notion that recall-oriented information seeking activity spans multiple cycles of interaction with the system is rooted in early research in library and information science (e.g., [4, 5, 6, 14, 15]). The notion that the query history should be represented in search systems dates back to at least the 1970s with systems such as DIALOG (see [22] for an example) that kept track of a searcher's queries and allowed those queries to be reused by reference. In the 1990s, web browsers quickly converged on the idea of using link color to reflect recent link traversal. A more modern example can be found in Ancestry.com [1], a commercial search engine for genealogical data that allows people to document family trees using historical records. It annotates search results with badges that show whether a particular record has already been associated with a person in the searcher's family tree.

These issues have also been explored in a range of research systems. VOIR [9] displayed the retrieval history of documents using histograms that represented rank information. Ariadne [21] created a visual representation of a search trajectory to review earlier actions. SearchPad [7] let people save and revisit queries and documents while conducting web search. Spoerri [20] showed overlap among search results submitted to different search engines, but these techniques could also be applied to queries in the same search task. Komolodi *et al.* [13] described a number of interface designs involving query histories after studying information seeking in the legal environment.

Reasons for including histories of interaction in information seeking interfaces include allowing searchers to review what has been done, and to try alternative formulations of queries to better approximate the latent information need. But there is more to history than just the list of queries and saved documents. NRT [19] implemented a more comprehensive history mechanism that recorded not only previously-run queries, but also the documents retrieved by them, making it possible for the searcher to scan the results list visually for new or for re-retrieved documents. Querium [10] implemented a principled framework for recording queries, and retrieved, viewed, opened, and saved documents to

keep track of searchers' entire information seeking activity as it spanned multiple engagements with the system. It then made this *process* metadata available to the searcher via faceted filters for restricting query results to select documents that were as yet unexplored, newly-retrieved, etc.

All of these systems look at various aspects of earlier search activity and make it possible to use that information in some way to help with sense making or query formulation. Auto-completion interfaces give searchers a preview of what they might want to look for through a query completion (e.g., [11, 3]) based on aggregations of prior queries collected from other searchers typing similar text. Google Instant [11] also shows the results of the most common expansion without having to press the Enter key to evaluate the query. Autosuggestion of keywords (e.g., [2, 12]) is another form of interactive query formulation assistance.

While query auto-completion is quite useful for precision-oriented, commonly occurring information needs, it does not translate as well to exploratory search in which a particular searcher's information need may be sufficiently different from those of others to make quality recommendation difficult. In situations that involve proprietary data (e.g., e-discovery, patentability searches, intelligence analysis, etc.), query histories to make accurate recommendations may not be available at all.

3. QUERY PREVIEW

One of the challenges of dealing with complex information needs through multiple queries is that queries can exhibit considerable overlap in terms of the documents retrieved. This makes it difficult for searchers to understand whether they are simply re-retrieving the same documents at different ranks, or whether they are in fact finding new information. These duplicates can impair a user's understanding of the search results and can interfere with an accurate sense of progress toward the search goal.

To address this problem, we developed a novel interface component to preview search results before the query is run. We wanted to bring the fluid style of interaction of Google Instant to the more complex expressions of information need typical of exploratory search. We designed a query preview widget that helps searchers understand what a query *will* retrieve before its results are seen. The system evaluates a searcher's query continuously as it is being typed (similar to Google Instant query completion), but rather than suggesting alternative queries, the system generates a visualization of the documents that would be retrieved if the query were executed, and contrasts these results with the documents that have been previously retrieved in the current search context.

The preview control is a stacked bar chart with ten bars. Each bar represents ten documents: the first bar represents documents ranked 1-10, the second represents documents 11-20, etc. Each bar is subdivided into three parts. These represent counts of documents that that will be: newly-retrieved (a bright teal blue), re-retrieved but have not yet been seen by the searcher (medium blue), and documents previously seen by the searcher (dark blue).

The goal of this control is to create a visual preview that indicates whether significant numbers of new documents will be retrieved by the query being constructed, and, if so, how these documents will be distributed throughout the overall ranked list. This is another important difference between the precision-oriented design of Google Instant and the recall-oriented design of Querium: here, the top 100 documents are represented, rather than just the top 10 shown in Google Instant.

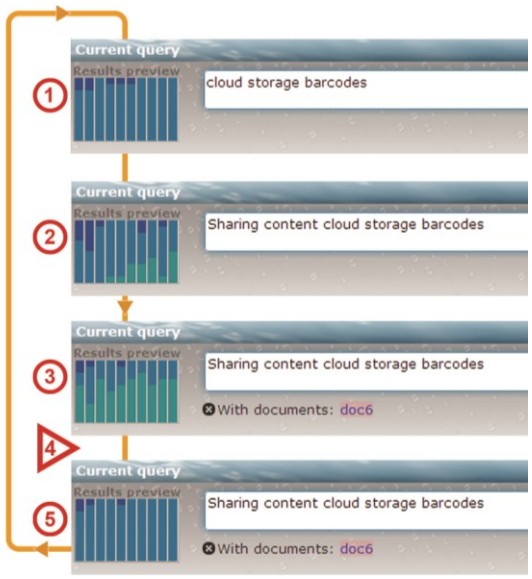

Figure 1. Example of the preview control as the searcher adds search terms (2), selects a document for relevance feedback (3), runs the query (4), and sees the final results (5)

Figure 1 shows changes in the bar colors representing the types of documents that will be retrieved by a query. Step 1 shows the control prior to modifying the query: some documents have been opened (dark blue, pages 1, 2, 4, 5, 6), while all others are marked as retrieved (medium blue) but not seen. As the searcher adds new search terms (step 2), the preview changes to reflect the types of documents the current query will return if submitted. The term "sharing content," will cause more previously-opened documents to be re-retrieved at higher ranks, but will also retrieve some new documents (teal) in the lower ranks. When the searcher adds a document as relevance feedback (doc6 in step 3), the preview control changes again to reflect that more new documents will be now be retrieved. When the searcher chooses to run the query (step 4) the preview control updates to show that all documents have either been opened or retrieved (step 5).

The preview is computed when the user pauses typing for about 300 msec, or when a document is selected or removed for relevance feedback [16]. Computation takes well under a second. Transitions between preview states are animated by adjusting bar component heights, producing an effect similar to spectral power displays in some audio equipment.

The design goal of this widget is to give searchers some insight into whether the query reformulation (e.g., adding a keyword or selecting documents for relevance feedback) will be effective at identifying new documents. It was also designed to increase the information scent [17] of documents in the lower parts of the result list, potentially giving incentive to explore the results in more depth. We also wanted to keep the interaction light-weight and modeless to avoid disrupting the searcher.

4. EVALUATION OF QUERY PREVIEW

To evaluate the impact of the preview on user behavior and search performance, we designed a study with two versions of an exploratory search user interface; one containing the preview control and one without. We used a simplified version of the Querium interface [10], described in more detail below, as the experimental interface. We were interested in testing three hypotheses related to recall-oriented search tasks:

Hypothesis One: The preview control affects searchers' attention and behavior during query formulation. People often look away while thinking [8], avoiding visual stimuli that may distract their cognitive processes; we wanted to assess whether people would be paying attention to the preview control as it was providing potentially useful information during query formulation.

Hypothesis Two: The preview control causes searchers to create queries that retrieve more different documents. Diversity of results is one key to more effective recall-oriented search. Would this control work as designed to increase the range of different documents people identify during a search task?

HypothesisT: The preview encourages deeper exploration of the search results. By definition, recall-oriented search relies less on the quality of the ranking function than precision-oriented search does. Would this control get people to look deeper?

4.1. Experimental design

The experiment was a one-factor within-subjects design. It compared two interface conditions, one with the preview, and one without (see Figure 2 and Figure 3), over a total of six different search topics (three in each condition). Topics were assigned to experimental conditions in a counter-balanced manner. Each participant performed three topics in each condition; each topic was performed once by each participant. Participants were randomly assigned to the counter-balanced configuration of topics, half starting with the preview condition and half starting with the control condition. The study was divided into two sessions, one for each condition usually run on separate days.

Figure 2. Query input area for the preview condition.

Figure 3. Query input area for the control condition

4.2. Search Topics

We designed experimental search topics to be as realistic as possible. Topics were framed as searches for relevant literature that might constitute prior or sufficiently-related art for proposed patent applications, a task requiring finding as many relevant documents as possible. In this context, not finding many relevant documents is not a bad outcome, as it may indicate that the research idea is a good candidate for a patent.

To construct the topics, we used summary descriptions of existing invention proposals and modified them to contain uniformly-detailed information and be of equal length. Some topics were updated to include modern technological components; overly technical terminology was simplified to allow us to include as many participants as possible and to leave room for query formulation. None of the participants in our study had specialized knowledge of the search topics, as these invention proposals had been created by people who did not participate in the study. However, participants had general knowledge about the research areas so that they could judge the relevance of the search results.

Each topic description included a title, a brief summary of the invention proposal (110-120 words), and eight keywords at the end of the description. A fragment of a topic description is shown in Figure 4 (5).

For each topic, we constructed two queries to seed the information seeking process: one query was derived from the topic title, and the other from three keywords in the topic description. Examples of two seed queries for a topic are shown in the query history in Figure 4 (4). Our motive, unknown to the participants, for providing the these two seed queries was to have some snippets to view when starting the topic, since this would allow the preview control to provide useful information from the participant's first query, and to focus their attention on query formulation. Participants were told that their partner had previously run a couple of queries to explore the search space. Since our participants often work with others on these kinds of tasks, picking up where someone left off was not unusual.

The following six topics were used in the study: "Text-reading support on handheld devices," "Creating movies of media streams on small devices," "Detecting and acting on multiple people crowding a small display for information sharing," "Improving interactions on mobile devices using large displays," "Sharing content using cloud storage and barcodes," and "Semi-Automatic Document Scanning with Digital and Video Cameras."

4.3. Search UI: Querium

We used Querium [10] as a platform to study the effects of the preview control in exploratory search tasks. Querium is an asynchronous collaborative search tool that organizes search activities into tasks; each task contains its own queries, retrieved documents, comments, and assessments of relevance. Within each task, a searcher can run multiple queries, examine results, save documents, perform relevance feedback (RF), etc. Querium makes it possible to perform relevance feedback by checking one or more checkboxes next to document snippets in the results list, and re-running the search. Terms drawn from selected documents are used to expand the query [16].

For the purpose of these experiments, we connected Querium to a snapshot of the CiteSeer database of academic papers, containing about two million documents. CiteSeer automatically extracts metadata such as author, title, and date from the PDF or Postscript files that it crawls, and also extracts the full text of the document. We used a snapshot of the CiteSeer corpus (including text and metadata) from June 2012, and built our own index of this collection using Lucene.

The Querium interface was simplified to focus participants' search behavior on query formulation. The study UI (Figure 4) organizes the display into several regions: the query area, the search results, a query history, and the document display area. PDF documents were replaced with their extracted text because iBrowser, the browser instrumented to collect eye-tracking browser data used for the study, could not display PDFs.

4.4. Participants

Thirteen participants completed the study. As search topics required domain knowledge, we recruited researchers and other members of the technical staff of our company to participate in the study. They did not receive any additional compensation. Five participants had used the full version of Querium previously; one had received a tutorial on the full version of Querium, and seven participants had not used Querium previously. All participants were familiar with the kind of search task involved in the study since similar tasks are part of their job assignment. None of the participants was actively involved in the development of the preview or of Querium.

It is worth noting that the interface of the experimental version of Querium shared only a few characteristics with the full version used previously by some of our participants. Most novel features had been removed to simplify the interface and to make the experiment more interpretable. In addition, all participants were given a 15 minute introduction on the experimental system. We believe that prior experience with Querium did not give those participants a material advantage, and, as the experiment had a within-subject design, the differences, if any, would cancel out.

Participants rated the two versions of Querium to support their search activities equally well (preview: 5.2, SD=1.36 vs. control: 5.2, SD=1.42 on scale 1-7 where 1 was very bad and 7 very good), indicating that they were equally satisfied with both.

4.5. Procedure and Instructions

The study was divided into two sessions, one for each condition with three search topics in each. Both sessions followed the same procedure. First, participants were given an introduction to Querium, after which the eye tracker was calibrated and the calibration was tested. Participants were then shown one of the three topic descriptions before using Querium. They were encouraged to read the description carefully so that they would not need to refer to it frequently while working on the topic, although the task description was available onscreen during the study (Figure 4 (5)).

Participants were instructed to quickly review what was done by their colleague, and then to run additional queries to find pertinent documents. To focus participants' activity on query formulation rather than on document review, they were told that they did not necessarily need to read the documents, but only to mark interesting documents by pressing the "thumbs up" button (See Figure 4 (2)). They were encouraged to work on the task until they felt they had exhausted the search space, or for at most 15 minutes. The maximum time was set to avoid fatigue and to assure that each search topic, independent of order, got about the same exposure in the study. Participants pressed the "done" button to move to the next topic. They were allowed to take short breaks between topics. Remember that for these search topics, not finding any relevant documents was not considered a bad outcome, so participants were not under pressure to find a large volume of relevant documents.

After a participant had completed the three search topics, the calibration of the eye tracker was tested again and the person was asked to fill in a short post-test questionnaire. Each of the two study sessions lasted in total 30 minutes to one hour; the second session was generally faster as the participants were now familiar with Querium and could skip most of the introduction.

4.6. Data collection and analysis

4.6.1. Log analysis

Querium was instrumented to report all significant user events (e.g., running a query, selecting a relevance feedback document, clicking on links, etc.) to the server, which kept a detailed log of these events. The log contained all queries that were run, and all documents that were retrieved, viewed, and saved by storing the data into a relational database. These logs and database records were also used in the following analysis to characterize searchers' behavior and results.

4.6.2. Ground truth

We created a set of ground-truth documents for each topic by pooling participants' results, sampling documents from the pooled set, and assessing these sampled documents in terms of relevance

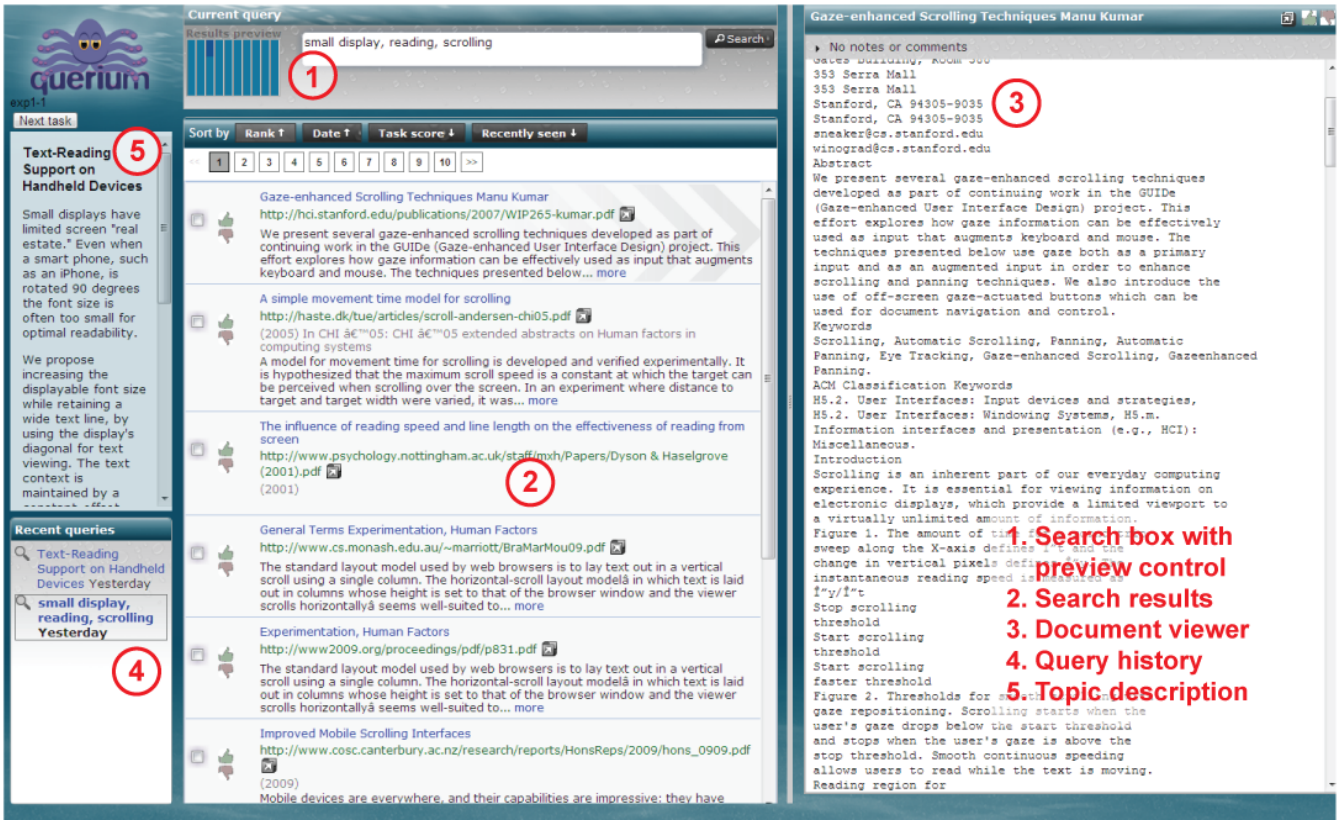

Figure 4. Querium interface for preview experiment.

to the task. Two assessors created the assessments independently. For each topic, each assessor rated 20 top-ranked documents. They then reviewed each other's decisions to arrive at a shared understanding of what constituted relevance for each topic. Finally, the assessors judged a random sample of documents drawn from the set of all documents at least one experimental participant had interacted with. In this context, "interacted with" means that the document was retrieved and either opened for reading, or the participant moved the mouse over its snippet.

Table 1. Total number of retrieved, assessed and relevant documents by topic

Topic	Retrieved in study	Assessed	Relevant
Topic 1	527	220	27
Topic 2	554	243	32
Topic 3	701	249	11
Topic 4	400	229	50
Topic 5	517	237	22
Topic 6	536	236	23

This set of documents judged relevant for each topic was used to score participants' performance in the experiment. Table 1 shows the total number of documents participants interacted with, the total number of documents assessed, and the total number of relevant documents for each topic.

4.6.3. Eye tracker

We recorded the participants' eye movements using a Tobii X120 eye tracker run at 60Hz. The gaze data was recorded by a custom piece of software (internally called iBrowser). iBrowser is a web browser that records eye tracker data synchronized with user initiated events such as key presses and mouse positions and clicks. iBrowser exposed a JavaScript API that the experimental Querium system used to communicate positions of UI elements as they changed based on searchers' interactions. This allowed us to track which controls and documents the searcher looked at relative to the logical structure of the interface during the search session. In addition to logging UI elements and eye tracking data, iBrowser also logged key presses and mouse interaction.

Fixations were identified in the eye-tracking data using a dispersion-based fixation detection algorithm [18]. When reporting on attention on different UI elements, attention data is based on the total fixation duration on that particular UI element. We used the following UI elements in the analysis: query (query input area excluding the preview and the search button), the preview control (when shown), search button, results, task description, query history, and document viewer.

We included only valid gaze data samples when calculating gaze durations. A valid gaze sample is when the eye tracker is correctly tracking at least one of the participant's eyes, and an invalid sample is when the eye tracker fails to track either eye. The average ratio of valid gaze samples was 0.82 (SD=0.094) over all conditions and topics. Only data samples with a ratio of 0.75 or higher valid gaze points for a specific time period (such as query formulation) were used in our analysis.

4.6.4. Data Analysis

For most of the analyses, unless noted, we used a one-way, repeated measures ANOVA with two conditions (control and preview). In cases where we expected that the participants would not display a consistent behavior, we used the t-test to compare the two conditions. This method was commonly used when analyzing queries and behavior during query formulation.

We used the ratio of valid gaze data samples as an indication of whether participants were looking at the display or away from it. This use of eye tracker accuracy is unconventional; other factors besides looking away from the display, such as rapid head movements, may cause loss of gaze data. However, we observed that many participants frequently looked away from the display as part of their natural movement pattern while interacting with Querium, (e.g., moving hand from mouse to keyboard, etc.). This loss of valid gaze data would be similar in both conditions, so any discrepancy in the ratio of valid gaze data samples in one condition over another could be assumed to arise from the participants looking away more frequently from the display.

When analyzing user behavior, we were particularly interested in participants' query formulation strategies. Using the mouse interaction and keyboard logs, we identified events representing query start (clicking in the search input area, adding a document to a query) and end (pressing the "enter" key or clicking the search button). Next, we analyzed fixations five seconds before the event, during the event, and five seconds after running the query.

For all analyses, we removed outliers that exceeded the mean by five standard deviations. The few points removed in this manner are reflected in the different numbers of degrees of freedom reported in tests of statistical significance.

5. RESULTS

The goal of this experiment was to characterize participants' behavior when using the preview control, and to understand its effect on overall system performance. We split the analysis into three parts: first, we characterize participants' behavior as observed through the eye tracker. Second, we describe the patterns of retrieval, viewing, and saving based on the data logged in Querium sessions. Finally, we compare participants' performance between conditions in terms of recall and precision.

5.1. Attention during query formulation

To test the first hypothesis, we examined participants' gaze patterns and behavior during the query formulation phase. The query formulation phase was initiated by activating the query box to type keywords or selecting documents for relevance feedback, and ended when the participant submitted the query. In the analysis, we included a five second time period *before* and *after* the query formulation phase to be able to compare behavior *before*, *during* and *after* query formulation.

Participants submitted on average 7.7 queries per topic in the control condition and 6.4 queries per topic in the preview condition ($F(1, 12) = 5.55$, $p < 0.05$). The time to formulate a query varied greatly, from 0.4 seconds to 7 minutes. The average query formulation duration was 21.4 seconds (SD=50.1) for the control condition and 27.2 seconds (SD=45.9) for the preview condition. Querium allows searchers to specify queries using a combination of keywords and documents for relevance feedback. We anticipated that the preview would be most useful for keyword queries (without documents) as these queries give searchers the most control over how a query is constructed. For these queries, the average query formulation duration was 12.4 sec (SD=20.1) for the control condition and 20.2 sec (SD=36.3) for the preview condition, which makes the query formulation on average 7.8 sec longer in the preview condition, a borderline statistically significant difference ($F(1, 12) = 4.43$, $p = 0.057$).

To investigate participants' attention during query formulation, we first examined the ratio of valid gaze samples in our data. We

found that this ratio of valid gaze samples was different before, during and after query formulation ($F(2, 22) = 40.13$, $p < 0.001$), and we saw an interaction between the time periods and the conditions ($F(2, 24) = 7.31$, $p < 0.01$). Further analysis showed that the time period *after* query formulation had a significantly higher ratio of valid gaze samples compared to gaze samples collected *during* query formulation (During vs. After: $F(1, 12) = 44.69$, $p < 0.001$, Bonferroni adjusted for multiple comparisons), as shown in Figure 5. The time period *before* query formulation was not quite significant (Before vs. During: $F(1, 12) = 6.47$, $p < 0.0774$, Bonferroni adjusted). The *before* and *after* time periods ratio did not significantly differ. A possible explanation of this curious result is that participants looked away from the stimuli rich display to collect their thoughts during query formulation, a behavior observed by the experimenter and consistent with research on gaze behavior during cognitively demanding tasks [8].

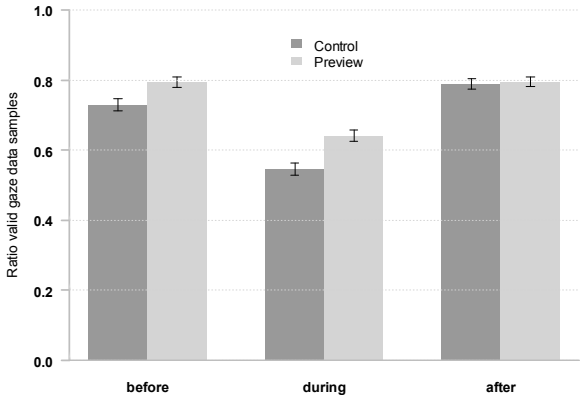

Figure 5. Ratio of valid gaze samples on the query area before, during and after query formulation.

The difference in the ratio of valid gaze samples during query formulation for the two conditions was significant ($F(1,12) = 8.18$, $p < 0.05$). These results show that participants looked at the display significantly more during query formulation when the preview control was available than in the control condition.

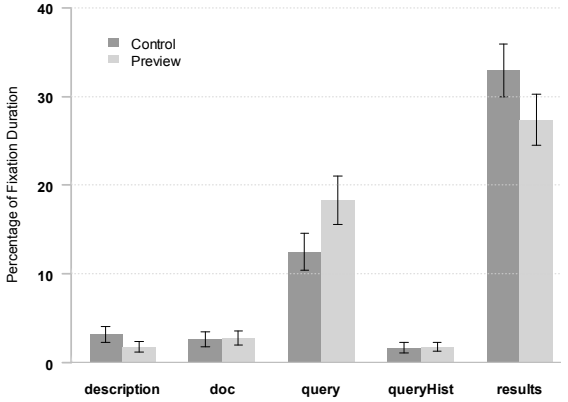

Figure 6. Percentage of attention on UI elements *during* query formulation (total fixation duration on UI element).

To further test the hypothesis that the preview control affects the participants' attention and behavior, we investigated the UI regions that participants looked at when formulating queries. Since the duration of query formulation varied considerably, we used the percentage of the total fixation duration during query formulation that was spent on the five major parts of Querium UI as the dependent variable (Figure 6). The biggest difference in attention is clearly related to the query input area; the second largest difference is in the results area. The difference in percentage of fixation duration on the query input area was statistically significant (Wilcox Rank Sum Test; W=3767.5, $p<0.05$). In absolute numbers, participants spent 6.1 sec (SD=8.08) looking at the query input area in the preview condition vs. 3.5 sec (SD=5.27) in the control condition. This result suggests that participants appeared to spend more effort at formulating queries in this condition compared to the control.

In the preview condition, in addition to spending 28% (SD=61.8) of the query formulation duration looking at the query area, participants spent on average 4% (SD=11.2) of the query formulation duration inspecting the preview control. We also found that participants spent on average 8% (SD=17.3) of the time period *before* starting on a query looking at the preview control and 7% (SD=15.3) *after* submitting the query. This corresponds to an average total fixation time of 391 ms (SD=865) looking at the preview *before*, 1021 ms (SD=2224) *during* query formulation and 362 ms (SD=764) *after* query formulation. The average fixation duration, when fixations were found on the preview control, was 297 ms (SD=139) which was not different from the average fixation duration on the query box (327 ms, SD=256). Considering the extra 7.8 seconds for query formulation in the preview condition, the ratio of that extra time spent on inspecting the preview is quite low. The additional time used for query formulation in the preview condition was more likely spent on formulating the query than interpreting the preview.

To understand the extra time participants spent on formulating queries in the preview condition, we investigated whether the query length differed between the two conditions. However, we found that query lengths were essentially equal: in the control condition, queries contained on average 5.3 words (SD=3.06) vs. 5.5 words (SD=2.62) in the preview condition. We also investigated if the participants made more edits, i.e. deleting or replacing query terms, to their queries. We found that in the control condition participants made 5.4 edits per topic (SD=4.46) vs. 6.6 edits per topic (SD=4.39) in the experimental condition. Due to the sparse sample of edits, the difference between conditions were not significant ($F(1, 12)=1.63$, *ns*.).

We also looked at individual differences in how participants spent their attention during query formulation. Some of the participants appeared to look for a longer time at the preview control than others. Of the 13 participants, nine looked at the preview control for at least 8% of the time period before, during or after query formulation. Of these, four participants looked at the preview control for at least 6% of the time period during query formulation. We did not find any differences in the use of the preview due to previous experience with Querium.

One interesting observation from the analysis of attention on the query box before, during, and after query formulation was that participants continued to look at the query box and the preview control after submitting the query. In the control condition participants' attention was shifted towards the search results, where it was 9% higher compared to the preview ($F(1, 11)=5.32$, $p<0.05$). The experimenter observed that participants seemed to try to use the preview control as a tool for navigating to newly retrieved material by placing the mouse on the preview control to count the bars and remember the location after the preview was flushed when new documents were retrieved. This was confirmed by participants' comments.

5.2. Query overlap

One of the motivations for the design of Querium was the observation that exploratory search tasks often involve queries that retrieve many of the same documents as searchers struggle to represent their information needs. We wanted to quantify this phenomenon in our data to validate some of the assumptions that underlie the system design.

In an on-going information seeking task, results overlap can be measured in a number of ways: it's possible to compare the results of each query to the union of the results of all preceding queries to assess its contribution to the entire task. It's also possible to measure query-to-query differences only, emphasizing incremental gains. Of course it's also possible to blend the two by discounting documents retrieved longer in the past. For our initial analysis, we chose the two extremes: the global uniqueness count and the incremental uniqueness count. For this analysis, we categorized queries as being based on keywords only, or a combination of keywords and documents for relevance feedback.

Table 2. Average percent of new documents per query by query type (QT), query overlap measure (global & incremental uniqueness) and experimental condition.

Query overlap	Condition	QT: Keyword M	QT: Keyword SD	QT: Document (RF) M	QT: Document (RF) SD
Global	Control	52.5	31.6	33.8	28.9
	Preview	58.0	29.8	41.8	27.8
Incremental	Control	71.6	27.8	48.0	30.7
	Preview	73.7	27.5	52.4	28.8

We calculated global uniqueness for a query by computing the number of documents it retrieved that had not been found up to that point in the search task. Note that some of these documents would likely be re-retrieved by subsequent queries. The numbers range from about 34% to about 58% for global uniqueness, and from 48% to 74% for incremental uniqueness (Table 2). While the query type effect is wildly significant ($F(1, 536) = 45.609$, $p < 0.001$), this is not surprising: relevance feedback queries produced lower results because some of the documents these queries retrieved had been previously found by keyword queries, as it is not possible to run a document query without first retrieving a document through some other query.

The difference in global uniqueness due to experimental conditions was significant ($F(1, 536) = 7.918$, $p < 0.01$). But differences in incremental uniqueness were not. Thus we clearly demonstrated that the initial assumptions regarding query overlap for exploratory tasks were valid, and also found support for hypothesis two (that the experimental condition would have less overlap). We take up that hypothesis again later in the performance analysis section.

5.3. Interaction results

We used several dependent measures to assess the impact of the preview on user behavior. The preview was designed to encourage people to look deeper in the results lists. Thus we used the rank of the document with which searchers interacted as an indicator of depth of exploration of the results. To test hypothesis three (that the preview control encourages people to explore more of the result set), we looked at the number of queries run, and at the rates at which participants viewed, opened, and saved documents in each condition. The only reliable difference between conditions was the number of queries per topic (Table 3).

Table 3. Summary statistics per topic. *p < 0.05.

	Control M	Control SD	Preview M	Preview SD	Sig. Test F(1, 12)
Topic duration (min)	12.2	3.16	11.7	3.18	< 1
No. Queries	7.7	3.54	6.4	2.52	5.55 *
Retrieved docs	525	186	522	123	< 1
Viewed snippets	76.9	39.3	73.4	37.7	< 1
Open documents	5.4	6.21	4.4	5.75	< 1
Saved documents	5.6	5.28	6.4	5.28	< 1

To explore participants' behavior in more detail, we broke down the viewed, opened, and saved document distributions by retrieval rank. To make the data easier to understand and to discount slight changes in rank (since a difference of a few rank places is not very important in recall-oriented search), we binned ranks into groups of 10, corresponding to the pages of results displayed by the preview control. As Querium retrieves up to 100 documents per query, each set of search results was divided into ten bins.

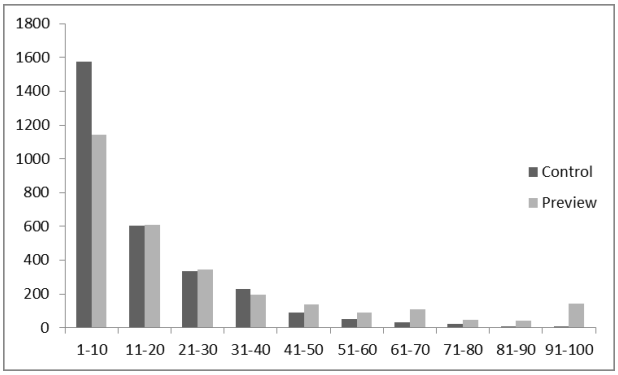

Figure 7. Distribution of viewed snippets by retrieval rank.

Figure 7 shows counts of ranks of viewed snippets for each interface condition. In the preview condition (light grey), participants examined many more documents at middle to lower ranks compared to the control. A Kruskal-Wallis test showed a significant effect of condition on viewed rank ($\chi^2(1)=132$, $p < 0.001$). This distribution suggests that in the preview condition participants devoted less attention to documents retrieved on the first page, and more to lower-ranked documents.

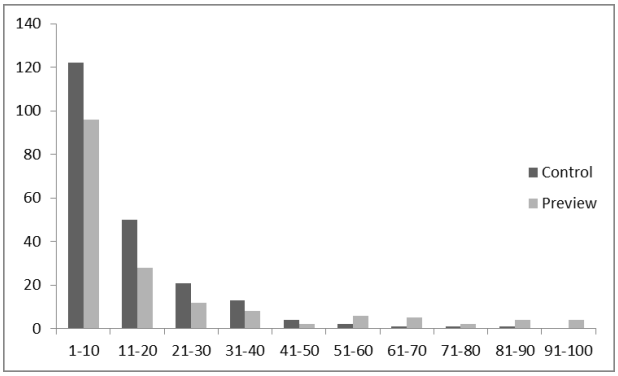

Figure 8. Distribution of opened documents by retrieval rank.

From eye tracking data we found that participants looked at each document snippet on average for 3.3 seconds in both conditions (Control: SD=20.53; Preview: SD=20.18). The number of unique documents per search topic that participants viewed for more than three seconds was 30.3 (SD=13.83) in the control condition, and

28.9 (SD=13.79) in the preview condition. These results show that participants allocated about the same resources to review the search results independent of condition.

We then compared the rates at which participants actually opened the documents to look at them, rather than relying on snippets alone. As can be seen from Figure 8, participants tended to open fewer documents in the preview condition in the top half of the ranks, and more in the bottom half. A Kruskal-Wallis test revealed a significant effect of condition on opened rank ($\chi^2(1)=4.0$, $p < 0.05$). While this effect is weaker than the viewed snippet rate discussed above, it does show a shift from opening documents from the top half of the ranked list to the lower half.

Finally, we examined the rate at which participants marked documents as being pertinent to their task. Pertinence was defined by participants' judgment rather than through an externally-imposed gold standard. The preview condition shows increased rates of documents being saved in the 11 to 100 rank range; a Kruskal-Wallis test revealed a significant effect of condition on saved rank ($\chi^2(1)=8.5$, $p < 0.001$), as shown in Figure 9.

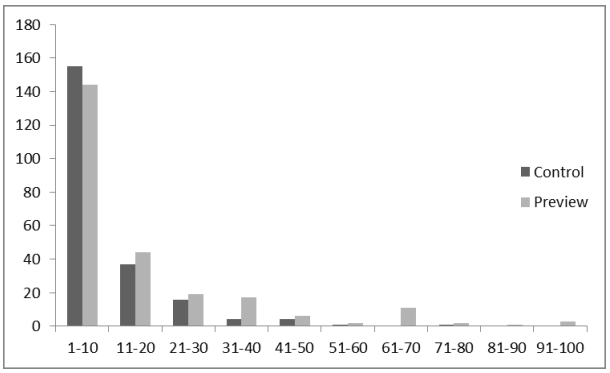

Figure 9. Distribution of saved documents by rank.

These analyses suggest that the preview encouraged participants to explore query results more thoroughly rather than running more queries. In addition, these explorations yielded more useful documents in the same amount of time as the control condition.

5.4. Retrieval performance

We assessed participants' performance by measuring residual recall (RR) and residual precision (RP) using the ground truth we had created. For each participant query in a topic session, we computed the number of *new* relevant documents retrieved, and used the presence of these documents to calculate RP and RR. Documents retrieved by the two seed queries in each topic were excluded from this analysis. Once a document was counted as being relevant to a query, it was not counted as being relevant when re-retrieved by subsequent queries within that topic. The goal was to measure how many new documents each subsequent query found, rather than simply re-retrieving the same documents.

The preview control was designed to facilitate deeper exploration of the results, a tactic we found our participants made use of in exploratory search. As can be seen in Figure 9, participants found and saved pertinent documents throughout the ranked lists returned by the queries. Thus to compare the experimental condition with the control using the ground truth, we wanted to compare the entire curve rather than just one or two points on it.

We computed the Average Residual Precision (ARP) metric by averaging RP computed at rank cutoffs of 10, 20, …, 100, and compared Mean ARP (MARP) between the experimental and control conditions. Three of 540 data points were excluded from RP analysis, and two were excluded from the analysis of RR because they were outliers.

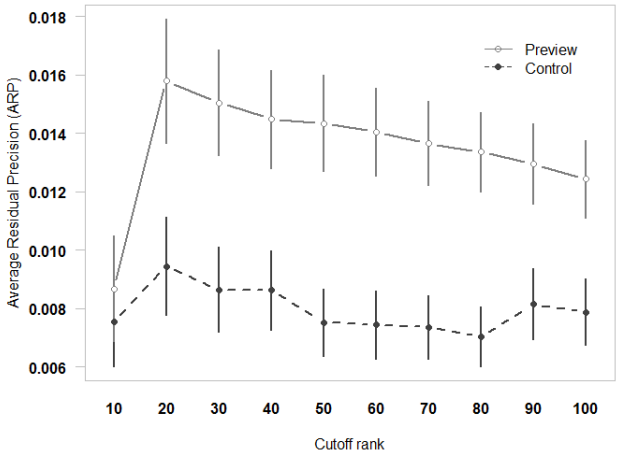

Figure 10. Average Residual Precision (ARP) vs. cutoff rank.

Figure 10 summarizes ARP over the cutoff range. Even without the statistical test, it is obvious that in the preview condition (upper curve), participants found significantly more relevant documents throughout ranks 20-100 compared with the control ($t(468.28)=3.553$, $p < 0.001$).

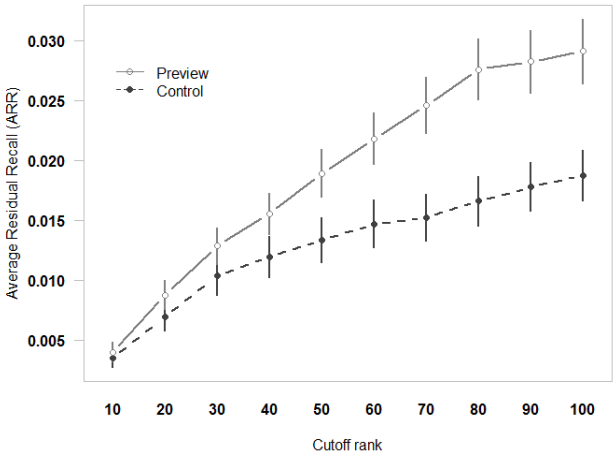

Figure 11. Average Residual Recall (ARR) vs. cutoff rank.

In an analogous manner, we computed Average Residual Recall (ARR), as shown in Figure 11. Here we see participants retrieving significantly more unique relevant documents over most of the ranked list ($t(536)=2.554$, $p < 0.01$). Thus not only was there no precision-recall tradeoff, but instead both RP and RR increased in when the participants had access to the preview control.

When comparing regular (not residual) recall and precision of the queries the two conditions, we find no statistically-significant differences (both $t(538) < 1$). We also find that the diversity of search results (the number of relevant unique documents retrieved per query) is significantly higher in the experimental condition (52 (SD=30.0) vs. 44 (SD=31.8), $t(538) = 2.7$, $p < 0.01$).

We conclude that in the control condition participants ran queries that tend to re-retrieve the same relevant documents, whereas in the preview condition they tended to retrieve a more diverse set of documents. This diversity caused an increase in the residual precision and recall rates.

6. DISCUSSION

We demonstrated that a small change in the interface produced a significant increase in the diversity, precision, and recall of search results in an interactive search environment without changes to the underlying search algorithms. Instead the system provided searchers with appropriate and timely feedback on the characteristics of the queries they were constructing, and gave them an opportunity to revise queries prior to viewing the results.

Interestingly, the tactics participants used to revise queries are hard to pin down. The average number of terms did not change; participants did not engage in trial-and-error tactics to select terms, as the average number of edits made to a query was about the same. Instead, it seems that improvements in retrieval performance were due to qualitative differences in how the queries were created. Our results suggest that participants put more thought into the search terms when the preview was present; they looked more at the user interface while formulating queries, and used on average 7.8 sec (29%) longer to formulate queries. This longer duration cannot be explained solely by participants looking at the preview. While participants looked at the preview for about one second on average, they looked about two and half seconds longer per query at the query input area. There seemed to be no advantage, however, to looking at other parts of the UI: participants spent more time looking at the results list, more time looking at documents, etc., in the control condition, but this did not result in better outcomes. This indicates that the preview control was working as we had intended, nudging people toward retrieving more diverse result sets and toward exploring these results more completely. It seems that people were engaged in more sense-making behavior both during query construction and when examining search results.

One effect of the search engine retrieving better results was that participants needed to submit fewer queries to find a satisfactory number of relevant documents. We saw a change in tactics in the experimental condition: participants looked deeper into the results lists, presumably due to the preview visualization. Participants tried to use the preview as a navigation tool, although it was not designed as such. As discussed earlier, examining lower ranked documents during exploratory search is important since the object is to find as many relevant document about a topic as possible rather than finding just one document. Residual precision and recall metrics underscore this as well: fewer unique documents were found in the ten top ranked documents than in the lower ranked documents. If these lower ranked documents had not been examined, they might not have been identified.

These results are encouraging. The design of the preview control is simple and requires little additional capability in the system; yet it prompted participants to formulate more effective queries. Our analyses of three independent aspects – gaze, interaction, and performance – all indicated that the preview had an effect on participants' behaviors and on outcomes. The consistency of these findings gives us some confidence that the effects are robust.

This work illustrates how interaction design can complement retrieval algorithm improvements. Empowering people's decision-making in complex search tasks can yield better outcomes for the combined human-computer system.

7. NEXT STEPS

In this study, we showed that the preview control helps searchers to make sense of the results a query is going to retrieve. Here we discuss how this control can be improved further.

7.1. Usability

While most of our participants understood the purpose of the preview, they described two main usability issues with the control: lack of visible change for some queries, and the resetting of the visualization when results are loaded.

The first problem occurs if the profile of the preview is identical to the profile of the current query. Participants perceived this lack of change as an error because they were biased to expect changes.

The second problem occurs when the preview resets after the results of the newly-run query are incorporated into the task workspace. While this is, in a sense, consistent behavior, it leads to poor usability due to a mismatch with user expectations. Also, resetting the display loses the opportunity to use the preview as a navigation mechanism to explore the new results, something that participants actively attempted during the experiment.

The design challenge is to represent these system states in a consistent, predictable, usable, and useful manner:

1. Query construction or reformulation: while it is working, the system should reflect that it is computing the preview.
2. Once the preview is computed, the system should indicate that the widget is displaying a preview. Even if the distribution of documents has not changed, the system should indicate clearly that it has recomputed the results.
3. After a query is run, the preview display should retain the previewed distribution and should act as a navigation mechanism into the document set retrieved by the query.

We are designing a new version of the control that preserves the visualization when the query is executed, and makes it possible to click on bars to navigate to corresponding pages of search results. The computation state is represented by a halo around the control; when results are available, the halo starts to pulsate gently, suggesting that the system is waiting for the searcher to react.

Figure 12. Mockup of alternative preview design.

Another possible design is to project the distributions of seen, unseen, and new documents directly onto the pagination controls, as shown in Figure 12. This solution decouples the preview from the display of retrieved documents. While this simplifies the individual widgets, it introduces additional complexity by representing related information in two different ways, and by complicating the design of the familiar pagination control. We are exploring this design space further.

7.2. Extensions in design

The preview control displayed three categories of information regarding documents that would be retrieved by the query being composed: whether or not the document had been retrieved previously, and if had been viewed or saved. It is possible to display other kinds of preview information as well.

Novelty could be defined in a fuzzy way based on significant rank promotion: when a document that has been previously retrieved at a low rank but has not been seen is retrieved at a significantly higher rank (e.g., a difference of 20 or more positions), its promotion could be indicated in the preview. This blurring of the

distinction between new and as-yet-unseen documents is probably a useful simplification but requires additional testing. Another variant could represent the amount of time that has passed since a document has been retrieved. This temporal expiration of whether a document has been found might be useful to remind people about early decisions in a long-standing information need. Time could be measured in absolute terms, or by including only the periods during which the searcher is interacting with the system.

7.3. Broader application

Querium was designed as an integrated information seeking environment that is, while web-based, a closed system. Yet some of the interactivity described in this paper can be applied beyond Querium to more generic search engines. One obvious application of the preview widget is a web browser plugin that monitors search activity and keeps track of found documents. The goal of this light-weight approach is to focus specifically on the task of managing the retrieval history in a nuanced and useful way. We are currently building a browser extension that tracks and visualizes re-retrieval patterns that occur during web search.

8. CONCLUSIONS

In this paper, we described a novel widget for helping searchers make sense of search results for complex information seeking tasks. We evaluated this widget in a controlled experiment to assess is impact on searchers' behavior. We found that it increases the rates at which participants examined documents at middle ranks in query results, and thus helped discover more useful documents in those middle ranks than without the preview widget. We also found that the preview control can increase the diversity of documents found in a search session, which can in turn lead to better performance in terms of recall and precision.

This exploration suggests that appropriately-designed interactive displays can be used to improve searchers' effectiveness in conducting searches for complex information needs. These kinds of visualizations use structural information collected during the search session to allow searchers to reason about the incremental result set. By making it easier for searchers to explore the results in more depth, we can reduce reliance on ranking algorithms that are only partially effective at predicting useful documents.

9. ACKNOWLEDGMENTS

We thank our participants for their efforts and Frank Shipman for some early discussions and insight. We thank Maribeth Back for her feedback on this paper.

10. REFERENCES

1. Ancestry.com http://www.ancestry.com
2. Bast, H., Majumdar, D., and Weber, I. (2007). Efficient interactive query expansion with complete search. In *Proc. CIKM '07*. ACM, New York, NY, USA, 857-860.
3. Bast, H. and Weber, I. (2006) Type less, find more: fast autocompletion search with a succinct index. In *Proc. SIGIR '06*. ACM, New York, NY, USA, 364-371.
4. Bates, M. (1989) The design of browsing and berrypicking techniques for the online search interface. *Online Review*, 13(5):407–424. Online: http://gseis.ucla.edu/faculty/bates/berrypicking.html
5. Belkin, N.J. (1980) Anomalous states of knowledge as a basis for information retrieval. *Canadian Journal of Information Science*, 5:133–143.
6. Belkin, N. J., Oddy, R. and Brooks, H. (1982) ASK for Information Retrieval. *Journal of Documentation*, 38, 61-71 (part 1) & 145-164 (part 2).
7. Bharat, K. (2000) SearchPad: Explicit Capture of Search Context to Support Web Search. In *Proc. WWW2000*, pp. 493-501.
8. Doherty-Sneddon, G., & Phelps, F. G. (2005) Gaze aversion: A response to cognitive or social difficulty? *Memory & Cognition*, 33, 727-733.
9. Golovchinsky, G. (1997) Queries? Links? Is there a difference? In *Proc. CHI 1997*. ACM Press.
10. Golovchinsky, G., Diriye, A., and Dunnigan, T. (2012) The future is in the past: Designing for exploratory search. In *Proc. IIiX 2012* (Nijmegen, The Netherlands). ACM Press.
11. Google Instant. http://www.google.com/insidesearch/features/instant/about.html
12. Hoeber, O. and Yang, X.D. (2006). Interactive Web Information Retrieval Using WordBars. In Proc *WI '06*. IEEE Computer Society, Washington, DC, USA, 875-882.
13. Komlodi, A., Marchionini, G., and Soergel, D. (2007) Search history support for finding and using information: user interface design recommendations from a user study. *IP&M*, 43, 1 (Jan. 2007), 10-29
14. Kuhlthau, C. (1991) Inside the search process: Information seeking from the user's perspective. *JASIS* 42,5, 361-371.
15. Marchionini, G. (1995). *Information Seeking in Electronic Environments*. Cambridge University Press.
16. Pickens, J., Cooper, M., and Golovchinsky, G. (2010) Reverted Indexing for Feedback and Expansion. In Proc. *CIKM 2010*.
17. Pirolli, P. and S. K. Card. (1999) Information foraging. Psychological Review, 106, 643-675.
18. Salvucci, D. and Goldberg, J. (2000) Identifying Fixations and Saccades in Eye-Tracking Protocols. In Proceedings of *ETRA'10*. ACM, 2000, 283-290.
19. Sanderson, M. and van Rijsbergen, C.J. (1991) NRT: News Retrieval Tool. *Electronic Publishing*, vol. 4(4), pp. 205-217.
20. Spoerri, A. (2004) How Visual Query Tools Can Support Users Searching the Internet. In *Proc ICIV'04*, London, UK, July 14-16 2004.
21. Twidale, M. and Nichols, D. M. (1998) Designing interfaces to support collaboration in information retrieval. *Interacting with Computers* 10(2), pp. 177-193.
22. Walker, G. & Janes, J. (1993) *Online retrieval: A dialogue of theory and practice*. Libraries Unlimited: Englewood, CO. p. 100.