Interactive Information Seeking via Selective Application of Contextual Knowledge

Gene Golovchinsky
FX Palo Alto Laboratory, Inc.
3400 Hillview Ave, Bldg 4
Palo Alto, CA 94304
gene@fxpal.com

Jeremy Pickens
FX Palo Alto Laboratory, Inc.
3400 Hillview Ave, Bldg 4
Palo Alto, CA 94304
jeremy@fxpal.com

ABSTRACT
Exploratory search is a difficult activity that requires iterative interaction. This iterative process helps the searcher to understand and to refine the information need. It also generates a rich set of data that can be used effectively to reflect on what has been found (and found useful). While traditional information retrieval systems have focused on organizing the *data* that was retrieved, in this paper, we describe a systematic approach to organizing the metadata generated during the search *process*. We describe a framework for unifying transitions among various stages of exploratory search, and show how context from one stage can be applied to the next. The framework can be used both to describe existing information-seeking interactions, and as a means of generating novel ones. We illustrate the framework with examples from a session-based exploratory search system prototype.

Categories and Subject Descriptors
H.5.m [**Miscellaneous**]: Interaction framework

General Terms
Human Factors

Keywords
Information seeking, session-based search, HCIR

1. INTRODUCTION
Information seeking can be a challenging activity. While in some cases, a precision-oriented approach such as that adopted by major web search engines can be effective at identifying the "perfect" document, there are many other situations in which a recall-oriented or exploratory search is warranted.

In such cases—when the automated, "ballistic" approach is not effective—the burden shifts onto the user. The user responds to this situation by running multiple queries [16], examining many more documents, [2] and building up an understanding of the collection and of the evolving information need (*e.g.*, [4]).

IIiX 2010, August 18–21, 2010, New Brunswick, New Jersey, USA.

Unfortunately, the tools designed for precision-oriented, single-document retrieval are poorly suited to more exploratory activity. To support exploration, to make sense of the patterns latent in the retrieved documents, tools have to be effective at exposing the structure of retrieved information. Making structure apparent can help searchers make sense of the information that was retrieved, and to help them understand whether the information being sought is available in the given collection or not.

So how can we structure the retrieved information? We argue that there are at least two dimensions along which information can organized: one based on *data*, and the other based on *process*.

Information retrieval interaction structure based on retrieved data is inherent in the objects being retrieved, but the degree of elaboration varies. In traditional full-text information retrieval, much of the document structure is implicit. In other cases, richer metadata such as dates, author, subject keywords, etc. may be available to group and filter documents. The extreme case is represented by faceted, multi-attributed data that can be grouped, filtered, and used for pivoting to help the user identify the useful set of documents.

Information retrieval interaction structure based on process can also be useful in exploratory search: A history of queries, documents, search terms, and other objects created or identified during a search session can be used in a variety of ways to aid the information seeking process. Allowing the user to reflect on what queries were run and which documents were retrieved can foster understanding of the entire session rather than just one search result list at a time.

In this paper, we propose a taxonomy of information seeking interfaces that represents the way structure is presented to the user and then focus on session-level, process-oriented structure. We characterize the range of possible session-based interaction patterns to suggest novel ways of browsing and searching through retrieved results. We then describe SACK (Selective Application of Contextual Knowledge), our session-based search interface. Finally, we use a working SACK prototype to illustrate some of the possibilities inherent in session-based (rather than single query) interaction.

2. RELATED WORK
We can divide the relevant related literature into two broad categories: faceted search (see [38] for an overview), and session-based search. We are interested in the broad details of faceted search; that is in making it possible for people to retrieve, sort, filter and pivot using metadata associated with the objects being retrieved. Facets can be intrinsic characteristics of the data, or may be extracted from unstructured or semi-structured document objects. Once metadata is identified it can be used in the user interface and in automatic query expansion [5].

Related to faceted search is a class of information visualization tools such as LifeLines [28] that trace their origins to Starfield displays [1]. These tools allow the user to select and display structured data for decision-making and sense-making by integrating various aspects of the data into coherent visualizations. This kind of browsing and filtering, however, deals with the data and metadata, rather than with the search process itself.

The term "session-based search" refers to information seeking support systems that use data from multiple queries specified by a user to inform the computation of subsequent search results, or their presentation in the interface, to support that user's information seeking. We contrast this with query-at-a-time systems that isolate the response to each query from those that preceded it. Merely keeping track of prior queries and saved documents is not sufficient; the system must also perform some computation on the history. Early information retrieval interfaces such as Dialog, for example, allowed users to combine earlier queries using Boolean operators, thereby reusing information created earlier in a search session (cited in [14], chap 4). On the other hand, systems such as Google that may use click information or other forms of implicit relevance feedback [18] of multiple users to affect the rankings of subsequently-retrieved documents are not session-based because they aggregate behavioral information without regard to a specific user's information need.

InformationGrid [33] implemented a mechanism for saving documents from different queries, and may have had a means for initiating relevance feedback from multiple documents saved by different queries. Its design also appears to have had a view for the query history, although implementation examples did not include that view. It did not, however, have any means to recombine queries or results lists.

VOIR (Visualization of Information Retrieval) [12] was a session-based search interface. The system kept track of retrieved documents and offered the searcher an interactive histogram for each retrieved document. In addition, it displayed a two-dimensional plot that represented retrieved documents by their highest and average ranks. The histogram offered users a snapshot of the retrieval history of a document, effectively identifying recurring vs. newly-retrieved documents. A similar technique appears to have been used by Shen and Zhai [34], although they offered no illustration and only a brief description of this visualization. The two-dimensional plot in VOIR made it possible to re-combine results of multiple queries to select documents based on their pattern of retrieval rather than on their contents.

Cousins *et al.* [7] described DLITE, a graphical interface for managing queries and search results against a range of collections. DLITE saved a history of queries and documents, and allowed searchers to save useful documents, but did not support query reformulation, relevance feedback, or any aggregation, sorting, or filtering of queries or of documents.

Twidale and Nichols [39] motivated their work on collaborative information seeking by observing that collaboration requires participants to share not just the results of their work, but also their search process. They found that users were not able to articulate their search process when requesting help with failed searches, and that having artifacts that reflected their activity or the context of their information need improved communication with a reference librarian. Since exploratory search is often not solitary (*e.g.*, [9], [26]), a system that maintains an actionable history of interaction may also offer opportunities for transitions to collaboration, whether the motivating task was initially conceived as collaborative or not. Ariadne [39] captured a search session over time and allowed the user to backtrack to previously-run queries or previously-viewed documents. It did not provide any automatic or semi-automatic means to recombine the objects to create new expressions of the evolving information need.

Session-based search has been discussed in the research literature previously by Zhai and colleagues [34] [35] [33]. This body of work focuses on the use of prior session context to help disambiguate and expand subsequent queries. They have found, for example, that combining query terms from a few queries produced improvements in precision at 20 documents and in average precision, particularly in combination with pseudo-relevance feedback. These results are consistent with history-based query expansion implemented in VOIR [12] that allowed users to browse documents by automatically constructing queries based on their interaction history.

Zhai *et al.*'s work, related in important ways to personalized search (see Kelly and Teevan [19] for a review), mines a searcher's prior activity for search terms that may influence the ranking of retrieved documents. This prior activity, or context, can be drawn from a user's file system [37] or from queries or documents identified earlier in a search session [34]. Personalization can be an effective means of implementing query expansion, and represents one component of session-based search, albeit an implicitly-constructed session. It is important to note that it is neither necessary nor sufficient, however: it is possible to implement session-based search systems without any information provided by the user prior to the start of the session, and it is possible to use personalization without implementing a session-based search system.

While the systems described above allowed users some ability to understand and react to their search history, they did not explore the range of possible interactions in a systematic way. In this paper, we offer a more comprehensive characterization of history-based interaction, and illustrate it with some examples from an information seeking support system we are developing. Our approach is based on characterizing objects identified or created during a search session. We analyze users' information seeking processes or actions in terms of the objects involved, and offer them means to examine, revisit, and recombine these objects in a variety of ways. Our interactive information retrieval system therefore builds an explicit representation of the users' own search processes as the context in which information seeking takes place.

The SearchBar system by Morris *et al.* [27] allows users to capture their web queries, browsing histories, notes and ratings. This session-based information is contextually interrelated, so that (for example) browsing history is tied to the queries (and search engines) that uncovered web pages in the first place. The purpose of retaining these relationships is to allow users to more easily re-find information, to resume and reacquire focus after an interrupted information seeking session or task. Their surveys show that there is a strong user need for session-based information seeking, with 83% of respondents engaging in multi-session (interrupted) seeking tasks. This work is similar to ours in that it allows users to return to previously executed queries and saved documents. However, there are two key differences. First, the SearchBar interface only stores information seeking activities hierarchically; the user can browse from Session to Query to Document, but cannot pivot to a document-centric view to get an overview of all the queries that retrieved a document. It also does not allow one to selectively reapply pieces of one's process-oriented context to find new information, a concept that we go into in Section 4.

From/to	Document	Query	Terms & metadata	Document Set	Query Set	User
Document	Link following, Related product recommendation	Query expansion/ relevance feedback	Show term-based summary of a document	"More like this" relevance feedback	Query expansion	Expertise finding, social search
Query	"I am feeling lucky"	Query term suggestion using pseudo-relevance feedback	Show terms that make up a query	Traditional IR	Yahoo! /Google related queries	Expertise finding, social search
Terms and metadata	Find document through term(s) or metadata	Use terms (with operators) to construct a query	Offer query expansion suggestions, correlated terms [6], faceted browsing	Filter list by included or excluded terms	Identify query suggestions based on specified terms	Expertise finding
Document Set	Personalized recommendation	Query expansion/ Relevance feedback	Shows terms common to multiple documents	Personalized recommendation, multi-document relevance feedback	Query expansion	Expertise finding, social search
Query Set	Identifying a promising document based on evidence from multiple queries (metasearch)	Identify a single query (either from or external to) that best exemplifies the set	Show terms that characterize the given query set, combine terms from multiple queries	Identifying promising documents based on evidence from multiple queries. (metasearch)	Query selection, query clustering	Expertise finding, social search
User	Document recommendations	Query recommendations	Expansion term recommendations	Document recommendations	Query recommendations	Social graphs queries

Table 1. Matrix of transitions among entities involved in information seeking. Each transition represents an opportunity to preserve or transform context.

While not necessarily exhaustive in its details, the approach we describe below represents a framework for handling historical process data that can be extended to additional types of data as appropriate to other retrieval situations.

3. THE FRAMEWORK

During the course of an exploratory information seeking session (a session that can span hours or even days), a searcher may run multiple queries, and find, examine, read, and save many documents. Exploratory search is often characterized by an evolving information need and the likelihood that the information sought is distributed across multiple documents. Thus the goal of the search process is not to formulate the perfect query or to find the ideal document, but to collect information through a variety of means, and to combine the discovered information to achieve a coherent understanding of some topic.

This piece-meal approach to information seeking, characterized as "berrypicking" by Bates [3], benefits from correspondingly flexible interaction. Our framework attempts to situate the behaviors described by Bates in terms of system and interface objects and user interactions with that system. The goal here is not only to offer a systematic treatment of HCIR [23] objects, but also to describe how context flows across them. The more that the various tactics identified by Bates and others can leverage each other, that is, to provide context for subsequent tactics, the more effective information seeking can be. Context can be carried among queries, documents, users, etc., and various such transformations have been implemented in many existing information retrieval systems.

Table 1 shows examples of transitions among the various objects that characterize a search session. These elements are the primary set of objects related to information seeking interfaces. Many examples correspond to existing, well-known information retrieval techniques, but some also suggest novel opportunities for interaction. Each transition is also an opportunity to preserve context, thereby enabling smoother, more effective interaction. Some sample transitions from existing work and from our prototype system are shown in Table 2.

Our framework emphasizes the variety of possible methods of expressing a user's information need. We argue that the different means of querying represented by this framework are useful for two reasons: first, individual differences among searchers may be reflected in different styles of information seeking, with some people preferring to type queries, others to follow links, etc. [12]. Systems that are designed to support multiple cognitive styles may therefore serve more people's needs better.

The second reason why a range of possible ways to articulate information needs is important relates to the evolving nature of the search task. As Kuhlthau [21] and Vakkari [40] point out, users' understanding of their information needs evolve over the course of a session. Early in exploratory search sessions, people tend to issue broad queries to gain a better understanding of the topic; later on, they often pursue more specific aspects with high-precision queries. In our framework, document- and document set-based queries may be appropriate at the earlier stages before users identify specific vocabulary that characterizes their information needs effectively. In later, more focused searches, queries consisting of specific terms may be employed. Toward the end, when reviewing and synthesizing information, cross-query document fusion can be effective for generating an overview of what was found over time, and may also suggest missing topics.

4. SACK: A PROTOTYPE SYSTEM

We built SACK (Selective Application of Contextual Knowledge), a prototype exploratory search system that explores some of the potential expressed in Table 1. For each explicitly-delineated search session, the system keeps track of users' information seeking activities, i.e. the entire context of an information seeking session. These activities include the queries each user issued, the documents (and their scores and ranks) retrieved for each query, and relevance judgments. By itself, keeping track of such information is not novel. Novelty arises in that SACK enables a user to pivot through this activity-based context to find, re-find, and discover information through selective application of that context. This is more than just allowing the user to examine the history of actions or saved information. The system gives the user the ability to selectively reapply arbitrarily-chosen information contexts, and to make *ad hoc* combinations of documents and queries to further explore the collection. SACK uses open source search engines (Lucene[1] and Terrier [30]) for its indexing and retrieval operations, and implements additional ranking algorithms.

Users interact with the system through a web-based interface, shown in Figure 1. The screen consists of a query entry field at the top, a results-browsing and pivoting area on the left, and the document display on the right. The browsing and pivoting area includes several tabs that partition the session objects into logical groupings, including query results, query expansion, query history, and document history. The tabs are synchronized, so that when a new query is run, the history updates; when a new relevance judgment is made, that judgment is reflected in every place where the document appears, etc. The user is free to explore and iterate the result set in any order, and can enter new queries at any time. Clicking on the down arrow next to a document displays a dialog through which judgments of relevance—useful, redundant, not useful, viewed, and not viewed—can be made.

SACK is designed to work with a variety of document types with different metadata. The system includes pluggable modules for different document types, but the process objects and associated pivoting code are generic. We have implemented two versions: one that works with TREC documents, and another with an internal collection served by DocuBrowse [11]. In addition to displaying document page images, DocuBrowse implements a feature that can highlight the specified terms on each page. We used that in conjunction with SACK queries to implement context propagation in the Query→Document and Query set→Document transitions. Unfortunately, we did not get a chance to implement the same highlighting feature for the TREC document collection that we discuss in the remainder of this paper.

We implemented a consistent and robust subset of the possible transitions to illustrate how search interaction can be enriched compared to conventional search interfaces. When designing or analyzing information seeking support tools, we must distinguish between users' perceptions of transitions and their implementation by a particular system. While the system may run or combine several queries and perform many computations in response to a user action, the user will still perceive it as a single action. Thus while some of the transitions we describe below are compounds from the perspective of building blocks of an information retrieval system, the interface we designed treats these operations as atomic.

4.1 Query→Document Set

SACK supports traditional, ad hoc retrieval; the system allows searchers to run full-text queries and displays search results in a ranked list. For each result, it shows document-level metadata, and allows the searcher to sort the results by any retrieved field. Clicking on the document title in the results list opens the document in an adjacent pane: the document is displayed in the context of the search results, rather than replacing them. Search terms from the query are highlighted in the document display to facilitate comprehension of the significance of the document

4.2 Document→Query Set

In addition to document-specific metadata such as dates, file size, etc., SACK displays a small histogram (Figures 1-6 and 8) for each retrieved document that shows the retrieval history of that document with respect to queries in the session. Each bar of the histogram represents the rank of that document with respect to each (time-ordered) query issued in the session: the taller the bar, the lower the rank of that document with respect to that query. At a glance, searchers can get a sense of how many queries retrieved a particular document, and at what relative ranks that document was retrieved by each. The visualization is active. Clicking on a bar will pivot the system to the results list of that query, showing the document in the context in which it was retrieved.

4.3 Document Set→Document Set

The system implements two transitions from a set of documents to another set of documents. After query results are retrieved, the user can mark some documents (currently, the values available are 'useful,' 'not useful,' 'useful but redundant,' 'viewed,' and 'not seen'), and request a related document set based on these judgments. The system will perform a relevance feedback operation on the selected documents to retrieve a new set of documents. However, rather than replacing the current query, the system creates and stores a new query object using the marked documents as the query representation. At any time the user can pivot back to the original (non-augmented) document set and mark or unmark additional documents, and run additional feedback queries. Conversely, the user can re-pivot to the augmented query and also mark or unmark additional documents and create yet another new query.

4.4 Query Set→Query Set

In the query history view the user is presented with a list of all queries that have been run in during the current session. Currently, this view allows the user to activate or deactivate any subset of these queries. This represents a transition in the use of context to inform subsequent search operations. Results of deactivated queries do not participate in the rank fusion that can be used to determine document order in the document history view. Thus the user can control which parts of the query history are used at any time to provide context for other search operations. This approach stands in contrast with the opaque use of search history by conventional search engines that may use prior queries to personalize search results (Google) [36], may show the recent search history (Bing), or may offer additional tools (Yahoo! Search Pad) [8]. These engines do not, however, let the user control which parts of the search history should be used and which should be ignored.

This is an important part of SACK, the selective application of contextual knowledge. Without such a feature, all queries in a session would be automatically added to a user's contextual information seeking activities. By giving the user control over exactly which historical actions are useful, the correct contexts may be applied at the correct stage in an information seeking session. Currently, the selection and de-selection of queries is a manual process. In the future, automated methods could be integrated, for

[1] http://lucene.apache.org/

From/To	Document	Query	Terms and metadata	Document set	Query set	User
Document	Relevant passage highlighting	Terms derived from a document are used to expand or generate a query	“Back of the book” index-style links	Terms derived from a document are used to identify matching passages in other documents	Strength of association between document and query can be used to reflect the query's rank	
Query	Query terms can be highlighted in the document	Queries can be reformulated based on query chain histories, often mined from log analysis [17]	Term clustering to show co-occurrence	Query-biased summaries can be generated for each document in a results set	Related queries can be partitioned based on different aspects of the original query	Users can be labeled by information needs implicit in the queries they issued
Terms and metadata	Terms can be highlighted in the document	User-selected terms can be used to expand queries	Term clustering to show co-occurrence	Documents can be grouped based on user-selected terms		
Document set	Terms from a document set can be used to annotate a newly-found document	Terms weights can be influenced by selected document passages	“Back of the book” index-style links	Recommender systems; set of documents are used to find similar sets	Terms weights can be influenced by selected document passages	
Query set	Documents can be annotated with terms from multiple queries. Query timeline for each document.		Term clustering to show co-occurrence	Query-biased summaries can be generated for each document in a results set	A query set can be used to find related sets of queries, or to add, remove, or replace query subsets	Same as query to user, but aggregated

Table 2. Some examples of context transfer across transitions.

example by offering system suggestions about subsets of queries to activate, deactivate, or replace. See section 5.1.6 for an illustration of query selection.

4.5 Query Set→Document Set

The document history view is an example of aggregating query results to produce a document set. The system collects results for all queries (or for a selected subset as per Section 4.4) by document and uses a histogram to display the document retrieval history for all documents that have been retrieved at least n times (where *n* is a user-settable parameter which defaults to two). While the view is similar to the search results view for a single query, there is one important difference: the list consists of documents contributed by multiple queries, and can be sorted based on cross-query, meta-search fusion scores [10] such as CombSUM, CombMNZ, freshness, or other measures of document importance. Documents can be viewed from this list, can be further assessed (for relevance, redundancy, etc.), and the searcher can pivot to a search results view for any document with respect to any query. When a document is viewed from this list, it receives highlighted terms from all queries that retrieved it, rather than from any individual query. Thus the document history view collects context over the entire session, whereas the search results view restricts it just to a single query.

5. SCENARIOS OF USE

It is difficult to evaluate this system using standard precision-recall TREC-like metrics. Certainly it would be possible to evaluate the precision and recall of any one component of the system: transitions from Query→Document Set (i.e. *ad hoc* retrieval), transitions from Query Set→Document Set (i.e. metasearch), or transitions from Document Set→Document Set (i.e. relevance feedback). However, transitions from Query Set→Query Set and Document→Query Set are more difficult to evaluate. Moreover, the perspicaciousness of the system as a whole arises not from any one transition, but through the ease by which any transition may be made and selective context may be moved across that transition. Furthermore, there are some conceptual problems in applying precision/recall-based evaluation to interactive systems because judgments of relevance on which the ground truth is based are rendered for each document in isolation, rather than for a growing set of documents collected throughout an interactive session. [15]

5.1 Details of interaction

Therefore, our evaluation in this paper will be example-based; we use a storyline and series of screen shots to illustrate the possibilities enabled by the system. This example, while illustrative in nature, is not a mockup; it is a genuine, unedited sequence of actions performed on the live system.

5.1.1 Step 0

In the initial step, the user selects a topic and explicitly creates a new session related to that topic. For this example, we chose TREC topic 354, on TREC disks 4 and 5 [40]:

> Title: journalist risks
>
> Description: Identify instances where a journalist has been put at risk (*e.g.*, killed, arrested or taken hostage) in the performance of his work.

5.1.2 Step 1

The user issues a query: [journalist risks], and then reads the top 10 documents in the result set, marking the documents at ranks 2, 3, 4, and 8 as relevant, and the remainder as non-relevant, as shown in Figure 1.

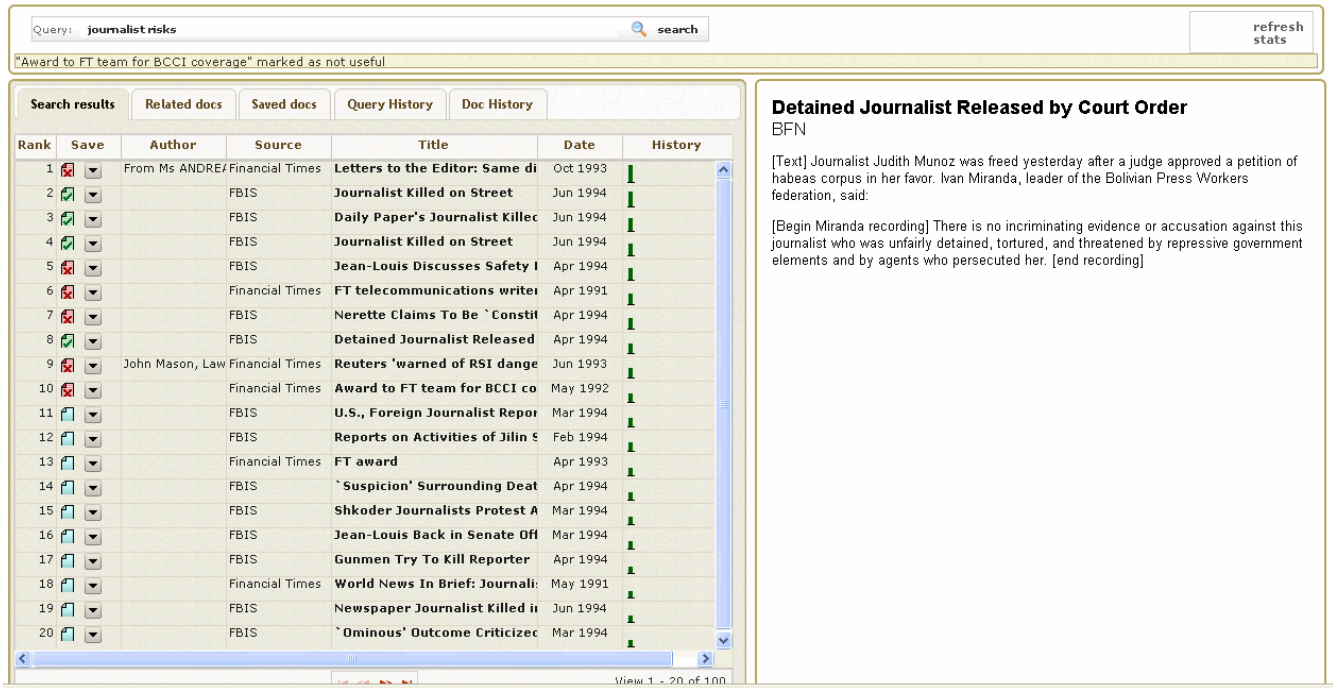

Figure 1. Initial query on the "journalist risks" TREC topic

5.1.3 Step 2

After reading the 8th-ranked (relevant) document the user gets an idea for additional keywords to use, ones that are a little more specific than "risks". So the user runs another query: [detain threaten torture journalist]. This query performs well, and the user reads and marks documents 1, 2, 3, 6, and 9 as relevant, and the remainder (up to the top 10) as non-relevant (Figure 2).

Figure 2. Second query, showing some overlap, but also some newly-retrieved documents

The user also notices from the Document→Query Set histogram next to each document in the results list that while there are a few documents that were retrieved by the previous query, many documents are new. This visualization makes it easier to assess the effectiveness of queries by visualizing their unique or repeated contributions to the set of retrieved documents.

5.1.4 Step 3

Again, based on terms found in one of the relevant documents from the previous query, the user issues another query: [journalist prison sentence]. Again it is clear from the Document→Query Set histograms in Figure 3 that the top ranked results that are quite different from the previous two queries. Documents are again marked for relevance (2, 4, 5, 7, 9, 10) and non-relevance.

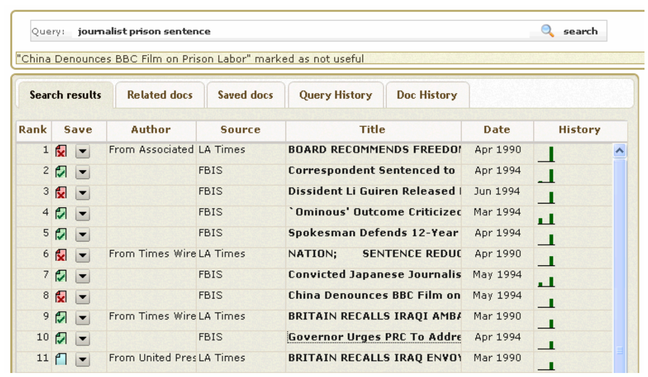

Figure 3. Another query, with some new and some re-retrieved documents, and more relevance judgments.

At this point, the user is out of ideas for new queries and decides to make a Document Set→Document Set transition by switching to the related documents tab, which runs a related documents query consisting of relevance feedback and query expansion from the previous query in Step 3. The user sees that this new action finds a lot of documents that have already been marked as relevant and non-relevant (Figure 4a). Furthermore, as one would expect, there is significant overlap (as observable in the histograms) between this and the previous query.

5.1.5 Step 4

However, there are also new documents that have not yet been judged: the very first unseen document at rank 3 is found to be relevant, as are documents at ranks 4, 5, 7 and 10. Histograms for some documents (*e.g.*, 14 and 15) indicate (Figure 4b) these newly-found relevant documents had already been retrieved by the some of the first three queries, but at low ranks.

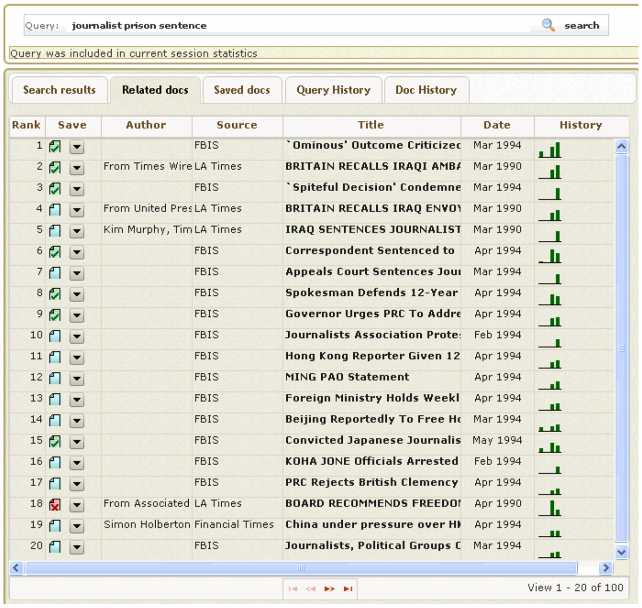

Figure 4a. Results of "Related docs" query before any new judgments of relevance.

5.1.6 Step 5

At this point, the user begins to suspect that there are more low-ranked useful documents to be found. The user switches to the Document History tab and performs a Query Set→Document Set transition. The documents in the Document History tab are sorted by a CombMNZ fusion score of the four queries from Steps 1-4.

The first three as-yet-unmarked shots are all relevant (Figure 5). While doing the relevance marking, the user notices that the queries that had retrieved these documents were {#1, #3, #4}, {#1, #2, #4}, and {#1, #3, #4}, respectively. That is, the first and the fourth queries were common to all of these newly-discovered relevant documents, but the second and third queries were not.

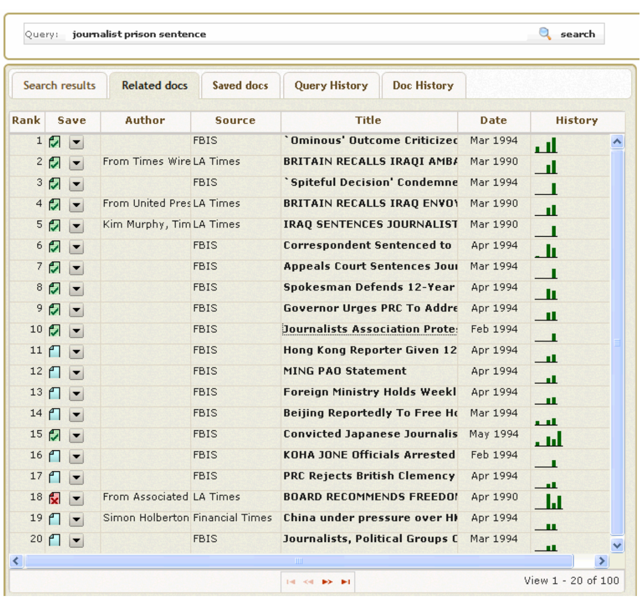

Figure 4b. More useful documents have been found and marked.

5.1.7 Step 6

Therefore, the user begins to suspect that a good avenue for exploration may be to combine only the first and fourth queries and not the middle two. The user transitions to the Query History tab and deselects these second two queries (Figure 6). Temporarily removing the queries from contributing to the document history is preferable to deleting them outright because the user can experiment with the effect of including or excluding specific queries without having to repeat searches.

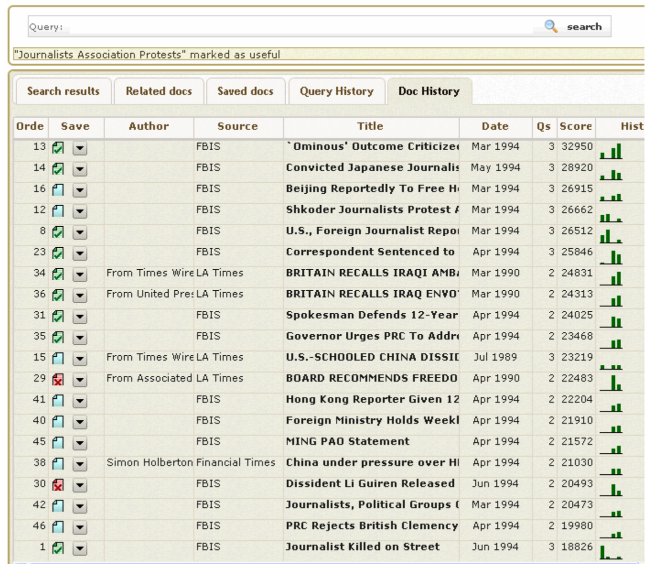

Figure 5. Document list from fusing the first four result sets.

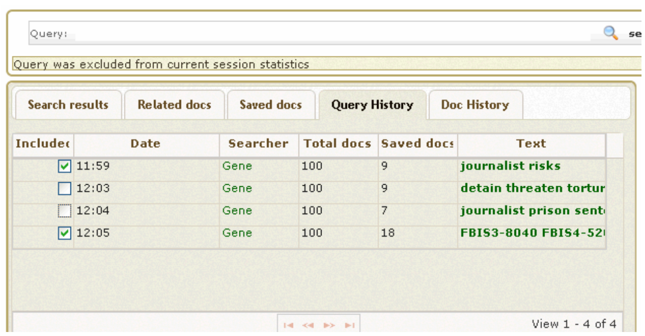

Figure 6. Removing the effect of two queries.

5.1.8 Step 7

The user then returns to the Document History tab (Figure 7). The de-selection in the previous step has the effect of removing from the CombMNZ fusion score the contribution of the middle two queries, thereby re-ordering the documents relative to Figure 5. And the very first unmarked document ("Editorial Upholds Verdict") is, again, relevant.

Recall again that each document is accompanied by a query histogram. These histograms are active, and a mouse-over reveals the ranks at which the document was retrieved by each previous query. We note that this newly-found relevant document was ranked 49th by query #1 and 30th by query #4. This document was also not found in the first page of results in the full query set fusion from Step 5. Rather, it is a document that revealed through selective application of process-based context (activating only queries #1 and #4) within the current information seeking session.

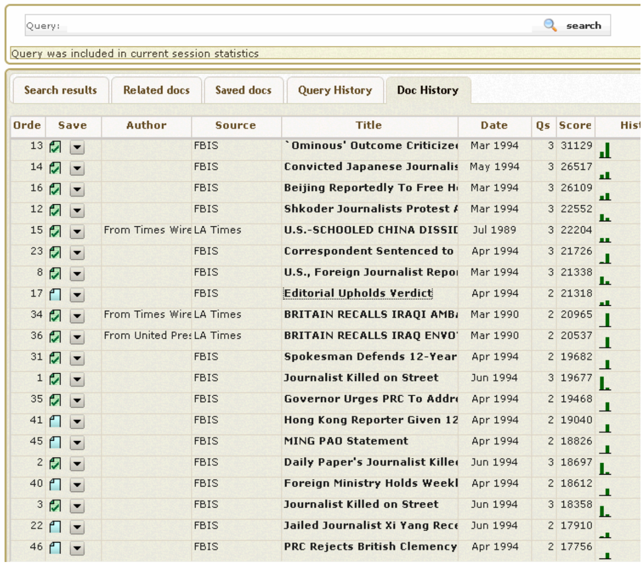

Figure 7. Documents identified only by Queries 1 and 4; Item #17 is discovered to be relevant.

6. Alternative scenario

The example-based evaluation in the previous section gives a taste of the type of activity and contextual flow that is enabled by this system. Certainly other pathways through one's session are possible. In fact, when constructing this example, we considered presenting another approach that we had taken. There were too many screenshots to concisely convey the flow of the system, so we will instead summarize the process.

First, we ran the same three queries and judged the same top 10 documents for relevance, as in Steps 1-3 in the previous section. Then, using the query history tab, we returned to the queries and performed a Related Documents relevance feedback step on each query, individually, in succession. While there was some overlap between document sets returned by each query, there was enough of a difference that the related documents discovered new sets of documents that were largely different, at least at top ranks. Feedback using the first query [journalist risks] produced half a dozen new relevant documents, feedback using the third query [journalist prison sentence] also produced half a dozen new relevant documents, but feedback using the second query [detain threaten torture journalist] produced a largely non-relevant set.

Pivoting into the document history view, thereby running fusion across all six queries (each initial query plus its augmented feedback version) produced a mixed bag of relevant and non-relevant results. So we returned to the query history view and deselected the middle query and its feedback counterpart. Even though there were relevant documents "contained" within those middle query sets, the remainder of the non-relevant documents veered sharply from the primary topic. After these queries were deactivated, we returned to the document history view to see the results of the remaining fused queries. Easily a dozen or more new relevant documents, ones that hadn't already been found even in the top ranks of the expanded queries, were immediately available. Furthermore, the query histogram accompanying each of these relevant documents shows that most of them were found across all four fused queries, but at originally-retrieved ranks around the 40s and 50s, with nothing better than the 30s.

6.1 Discussion

The fact that some of the relevant documents were promoted in rank by merging the results from several queries is to be expected from data fusion techniques. The point is not to reiterate the effectiveness of data fusion; rather, it is that the system allows the user to easily make decisions about when, which, and how to fuse existing queries, to run feedback, to run new queries, and to otherwise engage in session-based information seeking in a manner that allows preservation and alteration of one's activity-based context. This pivoting among queries and documents allows the user to understand the depth and breadth of search results and supports opportunistic strategies for discovering new information.

Vakkari [40] points out that "queries consisting of facets with several and perhaps specific terms have more discriminatory power than queries consisting of an intersection of facets…" The session-based approach we describe allows users to explore specific aspects of their information need with high-precision queries, while simultaneously providing an overview of a session that merges the contributions of the different queries.

When new documents are found, they are incorporated into the previously-identified set. The interface makes it easy to understand how much each new query contributes to the documents already found, whether the query overall is useful, redundant, or off-base. This continuous reflection is consistent with multi-query search strategies described in the literature (*e.g.*, [2], [16]) where multiple short queries are used to explore various facets of the search space.

While we did not focus on the particulars of any of the *ad hoc* ranking, results fusion, or query expansion step in this paper, we were able to show by example how a system that allows flexible and selective application of such tools can provide a rich, contextual, effective information seeking experience.

In this paper we hinted at evaluation by demonstrating effectiveness through an example series of interactions. The ongoing challenge for all session-based interactive search systems is to evaluate the effectiveness of not only the objective, algorithmic system capabilities, but the subjective and interactive possibilities that open up during the course of a session. This paper does not solve those issues, but the example interaction sequences that we have given demonstrate that selective application of context has significant potential.

6.2 Future work

SACK is by no means a complete system. One direction for continued work is to add support for additional transitions from Table 2, including Document set→Document set, Document set→Terms, and Terms→Document set. To this, we will add the capability to (optionally) select arbitrary sets of documents to decouple relevance feedback from the saving of documents, and we will expand its capability by adding an explicit way to deal with query terms (allowing the user to control query expansion).

Another direction for improvement is to apply some implicit relevance feedback techniques [18] to automatically mark documents as relevant if they match certain criteria. It will be interesting to test the relative effectiveness of explicit vs. implicit feedback: the rates of making decisions may trade off against the accuracy of inference.

It may be useful to show the chronological history of a session to users for purposes of reflection. Reviewing the moves that were made may trigger additional avenues of exploration, and may remind the searcher about alternatives that should be pursued.

Finally, we will explore how SACK supports collaborative search [31]. While the focus of the initial design was not explicitly-focused on collaborative search, we did build in multi-user capability into the system. Queries and assessments are associated with specific people (*e.g.*, Figure 7), and the system automatically combines inputs from multiple people just as it does from a single searcher. It may be useful to experiment with pivots that include or exclude specific users' contributions just as we handled specific queries in the example above.

6.3 Conclusion

Traditional exploratory search systems allow users to pivot around the data or information being searched, via such mechanisms as facets and clusters. Traditional session-based search systems store process information such as queries and relevance judgments. Our system, SACK, combines aspects of each, in that not only is a user's search activity stored, but the user may pivot around and recombine aspects of that activity to open up avenues of exploration into the data that might not otherwise have been discoverable.

7. REFERENCES

[1] Ahlberg, C. and Shneiderman, B. (1994) Visual information seeking: tight coupling of dynamic query filters with starfield displays. In *Proc. CHI '94*, ACM, New York, NY, 313-317.

[2] Aula, A., Khan, R.M and Guan, Z. (2010) How does search behavior change as search becomes more difficult? In *Proc. CHI 2010*, ACM, New York, NY, 35-44.

[3] Bates, M. (1989) The design of browsing and berry picking techniques for the online search interface. *Online Review*, 13:5, 407-431.

[4] Belkin, N. J. (1980). Anomalous states of knowledge as a basis for information retrieval. *Canadian Journal of Information Science*, vol. 5: 133-143.

[5] Ben-Yitzhak, O., Golbandi, N., Har'El, N., Lempel, R., Neumann, A., Ofek-Koifman, S., Sheinwald, D., Shekita, E., Sznajder, B., and Yogev, S. (2008) Beyond basic faceted search. In Proc. *WSDM '08*, ACM, New York, NY, 33-44

[6] Capra, R., Marchionini, G., Oh, J. S., Stutzman, F., and Zhang, Y. (2007) Effects of structure and interaction style on distinct search tasks. In *Proc. JCDL '07*. ACM, New York, NY, 442-451.

[7] Cousins, S.B. Paepcke, A., Winograd, T., Bier, E.A., and Pier, K. (1997) The digital library integrated task environment (DLITE). In *Proc. Digital Libraries*, ACM, New York, NY, 142-151.

[8] Donato, D., Bonchi, F., Chi, T., and Maarek, Y. (2010) Do you want to take notes?: identifying research missions in Yahoo! search pad. In Proc. WWW '10. ACM, New York, NY, 321-330.

[9] Fidel, R., Bruce, H., Pejtersen, A. M., Dumais, S., Grudin, J., and Poltrock, S. (2000) Collaborative information retrieval (CIR). *New Rev. Inf. Behav. Res.* 1, January (Jan. 2000), 235-247.

[10] Fox, E.A., and Shaw, J.A. (1994) Combination of multiple searches. In *Proc. TREC-2*. NIST Special Publication 500-215, pp. 243-252, 1994.

[11] Girgensohn, A., Shipman, F., Chen, F., and Wilcox, L. (2010) DocuBrowse: Faceted Searching, Browsing, and Recommendations in an Enterprise Context. In Proc. *IUI 2010*, ACM, New York, NY, 189-198.

[12] Golovchinsky, G. (1997) Queries? Links? Is there a difference? In *Proc. CHI '97*, ACM, New York, NY, 407-414.

[13] Harper, D. J. and Kelly, D. (2006) Contextual relevance feedback. In *Proc. IIiX 2006,* vol. 176. ACM, New York, NY, 129-137.

[14] Hearst, M. (2009) *Search User Interfaces*. Cambridge University Press.

[15] Ingwersen, P., Järvelin, K. (2005) *The turn: Integration of information seeking and retrieval in context*. Springer, Heidelberg, Germany.

[16] Järvelin, K. (2009) Explaining user performance in information retrieval: Challenges to IR evaluation. In *Proc. ICTIR 2009*, Springer-Verlag, Berlin, Heidelberg, 289-296.

[17] Jones, R. and Klinkner, K. L. (2008) Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs. In *Proc*. *CIKM '08,* ACM, New York, NY, 699-708.

[18] Kelly, D. and Belkin, N. J. 2001. Reading time, scrolling and interaction: exploring implicit sources of user preferences for relevance feedback. In *Proc. SIGIR '01*, ACM, New York, NY, 408-409.

[19] Kelly, D. and Teevan, J. (2003) Implicit feedback for inferring user preference: a bibliography. *SIGIR Forum* 37, 2 (Sep. 2003), 18-28

[20] Kriewel, S. Fuhr, N. (2010) Evaluation of an adaptive search suggestion system. In *Proc. ECIR 2010*, Springer, Berlin, 544-555.

[21] Kuhlthau, C. (1991) Inside the search process: information seeking from the user's perspective. *Journal of the American Society for Information Science*, 42(5), 361-371.

[22] Lu, J. and Zhou, M. X. (2009) An interactive, smart notepad for context-sensitive information seeking. In *Proc. IUI '09*. ACM, New York, NY, 127-136.

[23] Marchionini, G. (2006) Toward Human-Computer Information Retrieval. *ASIS&T Bulletin*. June/July 2006. Available online at http://www.asis.org/Bulletin/Jun-06/marchionini.html

[24] Marchionini, G., Brunk, B., Komlodi, A., Conrad, F., & Bosley, J. (2000a). Look before you click: A relation browser for Federal statistics websites. In *Proc. ASIS 2000,* 12-16.

[25] Marchionini, G., Geisler, G. & Brunk B. (2000b). Agileviews: A Human-Centered Framework for Interfaces to Information Spaces. In *Proc. ASIS 2000*, 271-280.

[26] Morris, M.R., (2008) A survey of collaborative Web search practices, In *Proc. CHI 2008*, ACM, New York, NY, 1657-1660.

[27] Morris, D., Morris, M.R., Venolia, G. (2008) SearchBar: A Search-Centric Web History for Task Resumption and Information Re-finding. In *Proc CHI 2008*, ACM, New York, NY, 1207-1216.

[28] Plaisant, C., Milash, B., Rose, A., Widoff, S., and Shneiderman, B. (1996) LifeLines: visualizing personal histories. In *Proc. CHI '96*. ACM, New York, NY, 221-227.

[29] Plaisant, C., Shneiderman, B., Muhslin, R. (1997) Information Architecture to Support the Visualization of Personal Histories. *Information Processing & Management*, 34(5): 581-597.

[30] Ounis,I. Amati,G. Plachouras,V. He,B. Macdonald,C. Lioma,C. (2006) Terrier: A High Performance and Scalable Information Retrieval Platform. In *Proc. SIGIR'06 Workshop on Open Source Information Retrieval (OSIR 2006)*; Seattle, Washington, USA. Available online at http://www.dcs.gla.ac.uk/publications/paperdetails.cfm?id=8265

[31] Pickens, J., Golovchinsky, G., Shah, C., Qvarfordt, P., and Back, M. (2008) Algorithmic mediation for collaborative exploratory search. In *Proc SIGIR '08*. ACM, New York, NY, 315-322.

[32] Rao, R., Card, S.K., Jellinek, H.D., Mackinlay, J.D., and Robertson, G.G. (1992) The information grid: A framework for building information retrieval and retrieval-centered applications. In *Proc. UIST 92*. ACM, New York, NY, 23-32.

[33] Shen, X., Tan, B., and Zhai, C. (2005) Context-sensitive information retrieval using implicit feedback. In *Proc. SIGIR '05,* ACM, New York, NY, 43-50.

[34] Shen, X. and Zhai, C. X. (2003). Exploiting query history for document ranking in interactive information retrieval. In *Proc. SIGIR '03*, ACM, New York, NY, 377-378.

[35] Sriram, S., Shen, X., and Zhai, C. (2004) A session-based search engine. In *Proc. SIGIR 2004*, ACM, New York, NY, 492-493.

[36] Sullivan, D. (2009) Google's Personalized Results: The "New Normal" That Deserves Extraordinary Attention. Search Engine Land. Available on the web at http://searchengineland.com/googles-personalized-results-the-new-normal-31290.

[37] Teevan, J., Dumais, S. T., and Horvitz, E. (2005) Personalizing search via automated analysis of interests and activities. In *Proc. SIGIR '05*, ACM, New York, NY 449-456.

[38] Tunkelang, D. (2009) *Faceted Search (Synthesis Lectures on Information Concepts, Retrieval, and Services)*. Morgan Claypool.

[39] Twidale M., and Nichols, D. (1998) Designing Interfaces to Support Collaboration in Information Retrieval. *Interacting with Computers*, 10(2):177–193.

[40] Vakkari, P. (2001) A Theory of the Task-based Information Retrieval Process: A Summary and Generalisation of a Longitudinal Study. *Journal of Documentation* 57(1): 44-60.

[41] Voorhees, E.M, and Harman, D. (1997) Overview of the Sixth Text REtrieval Conference (TREC-6). In *Proc. TREC-6*. NIST 1-24.