

By GARY MARCHIONINI

EXPLORATORY SEARCH: FROM FINDING TO UNDERSTANDING

Research tools critical for exploratory search success involve the creation of new interfaces that move the process beyond predictable fact retrieval.

F

rom the earliest days of computers, search has been a fundamental application that has driven research and development. For example, a paper published in the inaugural year of the *IBM Journal* 36 years ago outlined challenges of text retrieval that continue to the present [4]. Today's data storage and retrieval applications range from database systems that manage the bulk of the world's structured data to Web search engines that provide access to petabytes of text and multimedia data. As computers have become consumer products and the Internet has become a mass medium, searching the Web has become a daily activity for everyone from children to research scientists.

As people demand more of Web services, short queries typed into search boxes are not robust enough to meet all of their demands. In studies of early hypertext systems, we distinguished analytical search strategies that depend on a carefully planned series of queries posed with precise syntax from browsing strategies that depend on on-the-fly selections [7]. The Web has legitimized browsing strategies that depend on selection, navigation, and trial-and-error tactics, which in turn facilitate increasing expectations to use the Web as a source for learning and exploratory discovery. This overall trend toward more active engagement in the search process leads the research and development community to combine work in human-computer interaction (HCI) and information retrieval (IR). This article distinguishes exploratory search that blends querying and browsing strategies from retrieval that is best served by analytical strategies, and illustrates interactive IR practices and trends with examples from two user interfaces that support the full range of strategies.

Exploratory search. Search is a fundamental life activity. All organisms seek sustenance and propagation and Maslow's classic hierarchy of needs theory predicts that once people fulfill basic physiological needs, we seek to fulfill social and psychological needs to belong and to know our world. These higher-level needs are often informational and this in turn explains why information resources and communication facilities are so sophisticated in developed societies.

A hierarchy of information needs may also be defined that ranges from basic facts that guide short-term actions (for example, the predicted chance for rain today to decide whether to bring an umbrella) to networks of related concepts that help us understand phenomena or execute complex activities (for example, the relationships between bond prices and stock prices to manage a retirement portfolio) to complex networks of tacit and explicit knowledge that accretes as expertise over a lifetime (for example, the

most promising paths of investigation for the seasoned scholar or designer). For these respective layers of information needs, we can define kinds of information-seeking activities, each with associated strategies and tactics that might be supported with computational tools.

Figure 1 depicts three kinds of search activities that we label lookup, learn, and investigate; and highlights exploratory search as especially pertinent to the learn and investigate activities.¹ These activities are represented as overlapping clouds because people may engage in multiple kinds of search in parallel, and some activities may be embedded in others; for example,

lookup activities are often embedded in learn or investigate activities. The searcher views these activities as tasks, so we use "task" in the following discussion.

Lookup is the most basic kind of search task and has been the focus of development for database management systems and much of what Web search engines support. Lookup tasks return discrete and well-structured objects

such as numbers, names, short statements, or specific files of text or other media. Database management systems support fast and accurate data lookups in business and industry; in journalism, lookups are related to questions of who, when, and where as opposed to what, how, and why questions. In libraries, lookups have been called "known item" searches to distinguish them from subject or topical searches.

Most people think of lookup searches as "fact retrieval" or "question answering." In general, lookup tasks are suited to analytical search strategies that begin with carefully specified queries and yield precise results with minimal need for result set examination and item comparison. Clearly, lookup tasks have been among the most successful applications of computers and remain an active area of research and development. However, as the Web has become the information resource of first choice for information seekers, people expect it to serve other kinds of information needs and search engines must strive to provide services beyond lookup.

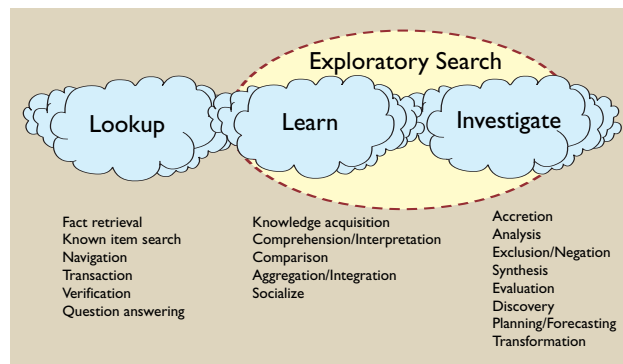


Figure 1. Search activities.

¹There are many important theoretical models of information search, for example, Saracevic summarizes Belkin's and Ingwersen's in his stratified model [9].

Searching to learn is increasingly viable as more primary materials go online. Learning searches involve multiple iterations and return sets of objects that require cognitive processing and interpretation. These objects may be instantiated in various media (graphs, or maps, texts, videos) and often require the information seeker to spend time scanning/viewing, comparing, and making qualitative judgments. Note that “learning” here is used in its general sense of developing new knowledge and thus includes self-directed life-long learning and professional learning as well as the usual directed learning in schools. Using terminology from Bloom’s taxonomy of educational objectives, searches that support learning aim to achieve: knowledge acquisition, comprehension of concepts or skills, interpretation of ideas, and comparisons or aggregations of data and concepts.

Another important kind of search that falls under the learn search activity is social searching where people aim to find communities of interest or discover new friends in social network systems (for example, www.friendster.com). Although the motivations may be distinct from other learning search examples, the exploratory strategies for locating, comparing, and assessing results are similar. Much of the search time in learning search tasks is devoted to examining and comparing results and reformulating queries to discover the boundaries of meaning for key concepts. Learning search tasks are best suited to combinations of browsing and analytical strategies, with lookup searches embedded to get one into the correct neighborhood for exploratory browsing.

Searches that support investigation involve multiple iterations that take place over perhaps very long periods of time and may return results that are critically assessed before being integrated into personal and professional knowledge bases. Investigative searches aim to achieve Bloom’s highest-level objectives such as analysis, synthesis, and evaluation and require substantial extant knowledge. Such searches often include explicit evaluative annotation that also becomes part of the search results. Investigative searching may be done to support planning and forecasting, or to transform existing data into new data or knowledge. In addition to finding new information, investigative searches may seek to discover gaps in knowledge (for example, “negative search” [1]) so that new research can begin or dead-end alleys can be avoided. Investigative searches also include alerting

service profiles that are periodically and automatically executed.

Serendipitous browsing that is done to stimulate analogical thinking is another kind of investigative search. Investigative searching is more concerned with recall (maximizing the number of possibly relevant objects that are retrieved) than precision (minimizing the number of possibly irrelevant objects that are retrieved) and thus not well supported by today’s Web search engines that are highly tuned toward precision in the first page of results. This explains why so many specialized search services are emerging to augment general search engines. Because experts typically know which information resources to use, they can formulate precise analytical queries but require sophisticated browsing services that also provide annotation and result manipulation tools.

These distinctions among different types of search activities suggest that lookup searches lend themselves to formalized turn-taking where the information seeker poses a query and the system does the retrieval and returns results. Thus, the human and system take turns in retrieving the best result. However, learning and investigative searching require strong human participation in a more continuous and exploratory process.

To support the full range of search activities, the IR community is turning increasingly to CHI developments to discover ways to bring humans more actively into the search process. Rather than viewing the search problem as matching queries and documents for the purpose of ranking, interactive IR views the search problem from the vantage of an active human with information *needs*, information *skills*, powerful digital library *resources* situated in global and locally connected *communities*—all of which *evolve* over time. The digital library resources are assumed to include dynamic contents such as other humans, sensors, and computational tools. In this view, the search system designer aims to bring people more directly into the search process through highly interactive user interfaces that continuously engage human control over the information seeking process. Although this is an ambitious design goal, we are beginning to see some progress in systems that are the forerunners to the exploratory search engines that will evolve in the years ahead.

TOWARD EXPLORATORY SEARCH SYSTEMS

Menus in restaurants serve the needs of both management and diners. From the system point of view, menus scope the kinds of products and services available and thus optimize performance; and from the patron’s point of view they simplify selection and

specification of gastronomical needs. In the computer industry, menus were the first kind of alternative to command systems and remain an important interaction style for selection and browsing. Expandable hierarchical file structures are specialized menus that serve as the mainstay of personal computing, cell phone, and PDA interfaces.

Hypertext links in texts were called “embedded menus” by Shneiderman [10] and current Web directory structures (for example, Open Directory) represent sophisticated menu structures for finding information on Web pages. In the database realm, query-by-example (QBE) interfaces were early alternatives to formal language interfaces and QBE-like systems remain the primary method for supporting non-textual queries in multimedia systems. These interface design experiences demonstrate the efficacy of selection as a form of query specification, and inspire link navigation as a primary user interface interaction style in the Web environment.

There is also substantial evidence in the IR literature that relevance feedback—asking information seekers to make relevance judgments about returned objects and then executing a revised query based on those judgments—is a powerful way to improve retrieval. However, practice shows that people are often unwilling to take the added step to provide feedback when the search paradigm is the classic turn-taking model. To engage people more fully in the search process and put them in continuous control, researchers are devising highly interactive user interfaces. Shneiderman and his colleagues created “dynamic query” interfaces [10] that use mouse actions

such as slider adjustments and brushing techniques to pose queries and client-side processing to immediately update displays to engage information seekers in the search process. A number of prototypes (for example, Dynamic Home Finder, SpotFire, TreeMaps) have come from these lines of research and development.

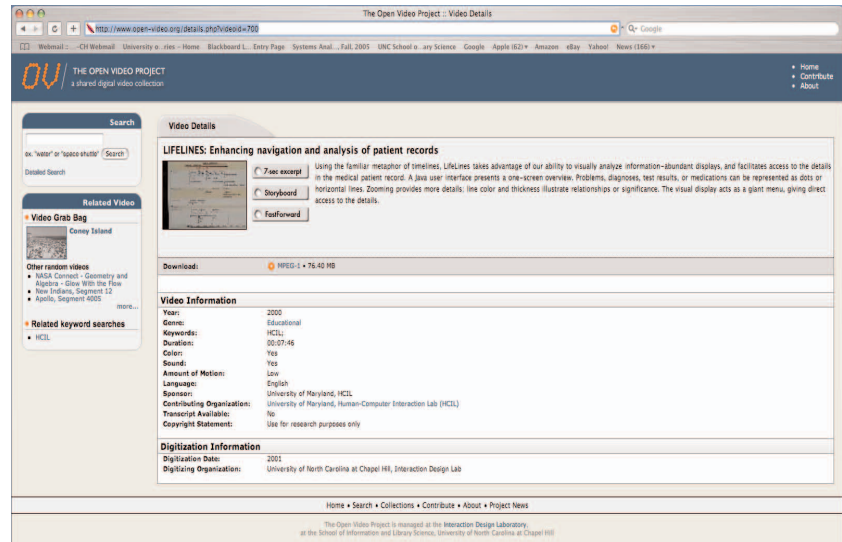


Figure 2. Open Video preview display for a specific video.

These techniques are especially good for exploration where high-level overviews of a collection and rapid previews of objects help people to understand data structures and infer relationships among concepts.

Other researchers have investigated these highly interactive interaction styles. Hearst and her colleagues created a series of interfaces that tightly couple queries to results, ranging from TileBars for text searching [2] to Flamenco (see the sidebar in this section), a series of interfaces that provides hierarchical, faceted metadata as entry points for exploration and selection. Hearst and Pederson [3], and others (for example, [11]) have used clustering of search results to make search more interactive, as represented by current Web search alternatives such as Clusty (clusty.com) that aim to provide groups of results that can be used to further search. Fox et al., schraefel et al., and Cutrell and Dumais offer other examples in



Exploratory search makes us all pioneers and adventurers in a new world of information riches awaiting discovery along with new pitfalls and costs.

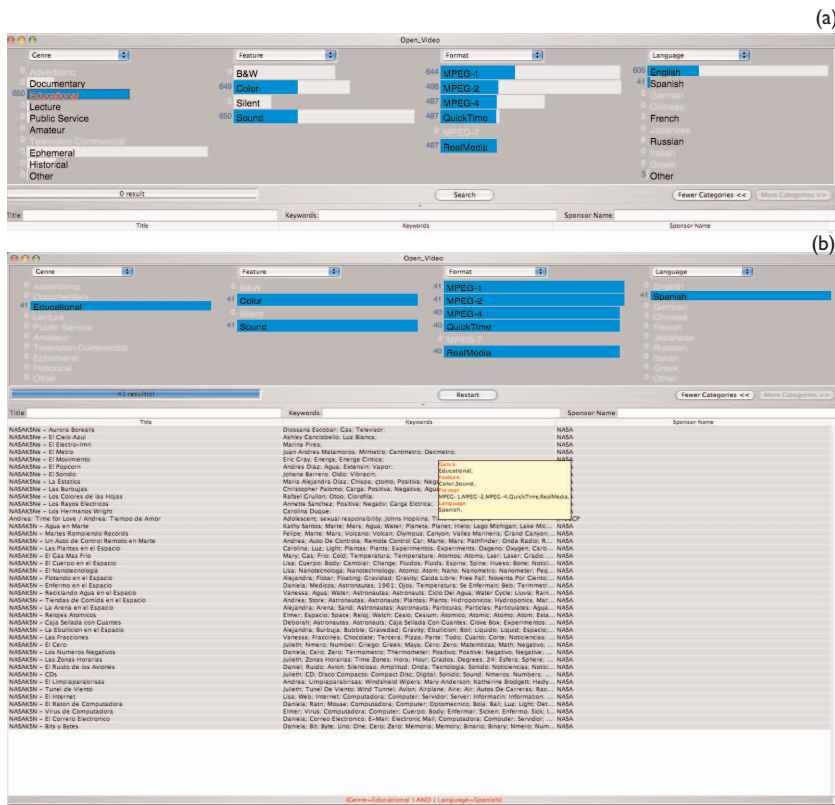


Figure 3. (a) Relation browser interface for Open Video Library with mouse over the education facet; (b) Relation browser display after educational and Spanish selected, mouse over fourth title.

for what is displayed (formats and level of text and visual detail) and how the results are ordered (relevance, title, duration, date, popularity).

Figure 2 shows a preview for a video with textual metadata and up to three kinds of visual surrogate (storyboard, fast forward, excerpt). The searcher may get more details by selecting the visual surrogate or download a video file in a format of their choice. The Open Video search system is meant to put people in control and support exploration as well as lookup. Our transaction logs indicate that half of the searches conducted begin with keyword strategies (analytical strategies) and the remainder begin with partition selection (browsing strategies).

this section of blending HCI and IR to support exploratory search. Our work at the University of North Carolina parallels these efforts and two example search systems that support exploratory search are illustrated here.

OPEN VIDEO EXAMPLES

The Open Video Digital Library (www.open-video.org) aims to give people agile views of digital video files [6]. The Web-based interface provides a number of alternative ways to slice and dice the video corpus so that people can see what is in the collection (overview) and determine greater details about a video segment (preview) before downloading it. There are different kinds of surrogates provided, including textual and visual representations and several layers of detail and alternative display options to give people good control. The user interface was designed to optimize agile exploration before downloading while allowing standard text-based search.

A number of user studies were conducted to determine which surrogates are effective and what parameters to use as defaults. This interface has proven to be quite effective over the past few years as thousands of users access the corpus each month to find videos for educational and research purposes. The home page provides a typical search form but also partitions the video collection in various ways so that people can select a specific partition to explore. Result set pages provide alternatives

As part of our efforts to develop highly interactive UIs that support exploratory search for government statistical Web sites, we developed a general-purpose interface called the Relation Browser (RB) that can be applied to a variety of data sets [5]. The RB aims to facilitate exploration of the relationships between (among) different data facets, display alternative partitions of the database with mouse actions, and serve as an alternative to existing search and navigation tools. RB provides searchers with a small number of facets such as topic, time, space, or data format; each of which is limited to a small number of attributes that will fit on the screen, simple mouse-brushing capabilities to explore relationships among the facets and attributes; and immediate results displays that dynamically change as brushing continues. Figure 3 illustrates how the RB works for a database such as the Open Video DL. Panel 3a depicts a portion of the RB at startup with the mouse positioned over the Educational category in the genre facet. The number of videos in the library in each of the facet-categories is immediately shown along with a set of bars that show the distribution visually. Thus, simply moving the mouse partitions the full corpus

into a view of the educational items. Clicking the mouse freezes this partition and allows continued browsing or retrieval of the partition from the server.

Panel 3b shows a portion of the display after the user has selected the Spanish language category within the educational partition and then clicked on the Search button. The display shows the number of items in each facet-category for the 41 videos in the result set in the upper panel and the titles, keywords, and producing agent for the videos in the bottom panel with additional metadata available on mouse-over. These items are hot linked to the Open Video DL. String search within the results fields is also supported and all results panel and query panel displays are coordinated to update in parallel when any mouse or keyboard action is executed.

The RB has been instantiated for dozens of databases, including several U.S. federal statistical agency Web sites. RB was designed to facilitate exploration and is less direct for simple lookup tasks than for exploratory tasks. Our user studies have demonstrated its efficacy when compared to standard Web-based retrieval. To support the dynamics, the metadata and query results must be available on the client side, thus limiting scalability to databases of roughly tens of thousands of items. We see this specialized kind of interface as an augmentation of today's powerful lookup engines. The RB could be used as a tool for exploring very large databases where the results are not individual items but subcollections or portals. Alternatively, the RB may be used after a standard Web search has been executed to investigate the result set if on-the-fly automatic classification is used.

CONCLUSION

It is clear that better tools to support exploratory searching are needed. Oblinger and Oblinger [8] argue the "Net generation" (those who learned to read after the Web) are qualitatively different in their informational behaviors and expectations; they multitask and expect their informational resources to be electronic and dynamic. The Net generation will expect to be able to use Web resources to conduct lookup, learning, and investigative tasks with fluid user interfaces.

As people spend more time online, not only will they increase their expectations about information tools and content, but there are more opportunities for mining their behavior patterns and applying adversarial computing that tries to take advantage of system and user behaviors. Exploratory search makes us all pioneers and adventurers in a new world of information riches awaiting discovery along with new pitfalls and costs.

Today, executing a query in a Web search engine not only returns results but targets the searcher for many kinds of presumably related opportunities and services. Exploratory search will exacerbate this trend as more user interaction data will be available for mining and analysis. One implication of considering good Web design that supports exploratory search together with client-side applications, like the RB, is to provide people with ways to trade off personal behavior data for added value services. Those who do not want their information behaviors to be mined can choose to use more client-side exploration tools, only sending requests for database partitions to the server.

Regardless of where the exploration takes place, it is clear that more computational resources will be devoted to exploratory search and the next search engine behemoths will be the ones that provide easy to apply exploratory search tools that help information seekers get beyond finding to understanding and use of information resources. **G**

REFERENCES

1. Garfield, E. When is a negative search result positive? *Essays of an Information Scientist 1* (Aug. 12, 1970), 117–118.
2. Hearst, M. TileBars: Visualization of term distribution information in full text information access. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems* (Denver, CO, 1995).
3. Hearst, M. and Pedersen, P. Reexamining the cluster hypothesis: Scatter/Gather on retrieval results. In *Proceedings of 19th Annual International ACM/SIGIR Conference* (Zurich, 1996).
4. Luhn, H.P. A statistical approach to mechanized encoding and searching of literary information *IBM J. of R&D 1*, 4 (1957), 309–317.
5. Marchionini, G. and Brunk, B. Toward a general relation browser: A GUI for information architects. *Journal of Digital Information 4*, 1 (2003); jodi.ecs.soton.ac.uk/Articles/v04/i01/Marchionini/.
6. Marchionini, G. and Geisler, G. The open video digital library. *dLib Mag.* 8, 12 (2002); www.dlib.org/dlib/december02/marchionini/12marchionini.html.
7. Marchionini, G. and Shneiderman, B. Finding facts vs. browsing knowledge in hypertext systems. *Computer 2*, 11 (Nov. 1988), 70–80.
8. Oblinger, D. and Oblinger, J. Is it age or IT: First steps toward understanding the Net generation. Educating the net generation. *Educause* (2005); www.educause.edu/educatingthenetgen.
9. Saracevic, T. The stratified model of information retrieval interaction: Extension and applications. In *Proceedings of the American Society for Information Science 34* (1997), 313–327.
10. Shneiderman, B. and Plaisant, C. *Designing the User Interface 4th Ed.* Person/Addison-Wesley, Boston, MA, 2005.
11. Zamir, O. and Etzioni, O. Grouper: A dynamic clustering interface to Web search results. In *Proceedings of WWW8*, (Toronto, Canada, 1999).

GARY MARCHIONINI (march@ils.unc.edu) is the Cary C. Boshamer Professor in the School of Information and Library Science at the University of North Carolina at Chapel Hill.

This work was supported by NSF grants IIS 0099638 and EIA 0131824.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

© 2006 ACM 0001-0782/06/0400 \$5.00