Searching with continuous query exploration

Sibel Adalı, Mousumi Chatterjee, Carrie Clegg, Manish Dalwadi, Ruchi Dayal, Jim Zappen
Rensselaer Polytechnic Institute

Teresa Harrison
University at Albany

We have observed three prevalent trends in interface designs for searching and browsing. One, search is considered as a one step process where the user, in order to locate items of interest, needs to modify the query if dissatisfied with the retrieved results. Two, browsing is usually implemented as a static process that does not relate to the previous searches. Three, adaptive interfaces for searching try to filter and push items to the user based on her past queries and interest. None of these approaches, independently, try to address the issue of searching and browsing as an integral whole which contains a set of interrelated queries with a complex need. A need to capture the users' past behavior preferences with the present motivations. We believe these processes are not inherently separable. They are integral and continuous. Therefore this research wants to capture the process of finding information through continuous information exploration and address the problem of managing interrelated queries for every individual from past as well as present behavior. Thus this system will enable its users to continuously explore and seek all the alternatives in a way such that the users get to explore these through a single interface, without having to move between screens or having to form search strategies to locate information while on one screen. The goal is to capture and adapt the experience for each individual, based on their need. It is then the duty of our explorer system to guide the users through the possibilities by either: (a) allowing the user to browse the objects, (b) allowing the user to specify new search constraints and assign ranking to the desirability of these constraints by observing user behavior, (c) requesting the user to provide feedback on the desirability of different options to refine the ranking process.

This browser capability is designed in a way to specifically show objects in a ranked order computed as a result of multiple searches. This ranking considers both, the desirability of the current options and the past user history. Furthermore, the browser is also designed to seek specific feedback from the user, implicit and explicit. Each continuous user query is implemented as a ranked relation with an associated rank column for each different method for ranking user preferences based on their queries [1]. The software agent uses this structure to make decisions on what to display and how, and when to ask for additional feedback. As an example, consider the case where a set of activities match a given search criteria. The system in this case has two alternatives for finding a search criteria, such that when specified, filters a large set of results, or results in many good matches. In the first case, the user is given options that separate the current search space to equal sized pieces. This allows the user to filter out many objects that are not of interest. In the second case, the user is given options that separate the search space into one large and possibly smaller pieces. This allows user to adapt his/her criteria towards what is available in the system whenever it is convenient for him/her. We show that both types of criteria can be defined as properties of the rank distribution of a cine query.

The notion of scores for ranking information has been investigated in the literature of information integration and meta-searching [3, 5]. However, scores are not always useful for reasoning with the explicit ordering of objects [2] and ranking objects based on score combination methods do not always satisfy some basic notions of fairness defined for voting systems [4]. Our ranked relation storage and query methods allows us to capture scores and ranking methods uniformly.

References

[1] Adalı, S., Bufi C. and Sapino, M.L. (2003) "Ranked Relations: Concepts, Applications and Query Languages for Multimedia", to appear in Int. Workshop on Multimedia Information Systems, 2003.

[2] P. Bonatti, M.L. Sapino and V.S. Subrahmanian, "Merging Heterogeneous Security Orderings", Journal of Computer Security, 5(1), pp. 3–29, 1997.

[3] Fagin, R. (1999) "Combining Fuzzy Information from Multiple Systems", JCCS, 58(1), pp.83–99.

[4] Dwork, C., Kumar, R., Naor M., Sivakumar, D. "Rank Aggregation Methods for the Web", in *Proceedings of the WWW Conference*, 2002.

[5] Ilyas, I., Aref, W., Elmagarmid, A. "Joining Ranked Inputs in Practice", in *Proceedings of VLDB*, 2002.