Task-Aware Search Assistant

Henry Feild & James Allan

Center for Intelligent Information Retrieval
Dept. of Computer Science
University of Massachusetts
Amherst, MA 01003
{hfeild,allan}@cs.umass.edu

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Miscellaneous

Keywords

Search task identification, search history

Extended abstract

When users search the web, their goal is to accomplish something, which we call a *search task*. Previous studies have observed an average of 2.6–3.3 queries per search task [2, 3, 4] and Jones and Klinkner [2] found 16% of tasks were revisited over a three-day span of Yahoo! logs. These statistics suggest that a task-aware search assistant may aid users in continuing previous search tasks as well as new ones.

There are several existing tools that are related or similar in nature to a task-aware search assistant, the most basic of which are built-in browser histories and extensions that make them easier to navigate. Both Google and Bing provide search histories over their respective services. However, these lack a task-oriented view of searches. The most closely related work that we are aware of is Yahoo's Search Pad, which attempts to automatically identify that a user is conducting research (e.g., planning a vacation). It then notifies the user and begins tracking queries and result clicks in an interface, to which the user can add and remove content. Two drawbacks are its lack of support for non-research tasks and its restriction to Yahoo! search only.

Our approach to aiding users during search—both for new and continuing tasks—is to present them with an easy to use sidebar implemented as a browser extension. The interface consists of a collapsible sidebar in the browser with multiple panes that can be paged through using streamlined navigation buttons. The primary pane displays three key groupings: (1) a list of the most recent search tasks, (2) a list of related queries entered in the past, and (3) a list of related tasks conducted in the past. Note that a task consists of a cluster of queries and visited documents. This primary pane is updated with each new query and page visit, adjusting for new tasks as they are detected. A second pane allows users to search and browse their entire task history as a chronologically ordered list. A third pane provides similar functionality, but without task clustering. We expect these functionalities to aid users in orienting themselves among their current and past tasks, and aid them in pulling in information from previous tasks, whether for refinding or for recalling vocabulary. As a browser extension, the interface can extract search tasks from the existing browser history, making the tool useful right out of the box. Unlike many of the search assistant tools describe above, this interface has access to searches from virtually all search engines and web pages visited at any point during a user's search—not just results clicked from a search results page.

The underlying technology relies on performing same-task classification between pairs of queries. Following previous work [1, 2], we use logistic regression models, which are well-suited for binary classification. We also use many of the same lexical features: Jaccard coefficient, tri-gram character overlap, and Levenshtein distance. We borrow a feature presented by Lucchese et al. [4], which computes the semantic similarity between two queries by the cosine similarity between vectors of tf-idf scores over Wikipedia documents. We also use their weighted connected components clustering method once the pair-wise classifications have been made.

In summary, our prototype provides a novel interface for providing users with information about current and past search tasks. The underlying task-classification is based on a fusion of existing methods.

Acknowledgments. This work was supported in part by the Center for Intelligent Information Retrieval and in part by NSF grant #IIS-0910884. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsors.

References

[1] P. Boldi, F. Bonchi, C. Castillo, D. Donato, A. Gionis, and S. Vigna. The query-flow graph: model and applications. In *Proc. of CIKM'08*, pages 609–618. ACM, 2008.

[2] R. Jones and K. Klinkner. Beyond the session timeout: automatic hierarchical segmentation of search topics in query logs. In *Proc. of CIKM'08*, pages 699–708. ACM, 2008.

[3] T. Lau and E. Horvitz. Patterns of search: analyzing and modeling web query refinement. In *Proc. of User Modeling'99*, pages 119–128. Springer-Verlag New York, Inc., 1999.

[4] C. Lucchese, S. Orlando, R. Perego, F. Silvestri, and G. Tolomei. Identifying task-based sessions in search engine query logs. In *Proc. of WSDM'11*, pages 277–286. ACM, 2011.

SIGIR'12, August 12–16, 2012, Portland, Oregon, USA.