New methods for creating testfiles: Tuning enterprise search with C-TEST

David Hawking,[1] Paul Thomas,[2] Tom Gedeon,[3] Timothy Jones,[3] Tom Rowlands[2]
[1]Funnelback [2]CSIRO [3]Australian National University
david.hawking@acm.org, paul.thomas@csiro.au, tom.gedeon@anu.edu.au, tim.jones@anu.edu.au, tom.rowlands@csiro.au

1. INTRODUCTION AND BACKGROUND

An evolving group of IR researchers based in Canberra, Australia has over the years tackled many IR evaluation issues. We have built and distributed collections for the TREC Web and Enterprise Tracks: VLC, VLC2, WT2g, WT10g, W3C, .GOV, .GOV2, and CERC. We have tackled evaluation problems in a range of scenarios: web search (topic research, topic distillation, homepage finding, named page finding), enterprise search (tuning for commercial purposes, key information resource finding and expertise finding), search for quality health information, automated bibliography generation, distributed information retrieval, personal metasearch and spam nullification.

We have found in-situ, in-context evaluations with real users using a side-by-side comparison tool [3] to be invaluable in A v. B (or even A v. B v. C) comparisons. When a uniform sample of a user population uses an n-panel search comparator instead of their regular search tool, we can be sure that the user needs considered in the evaluation are both real and representative and that judgments are made taking account the real utility of the answer sets. In this paradigm, users evaluate result sets rather than individual results in isolation.

But side-by-side comparisons have their drawbacks: they are inefficient when many systems must be compared and they are impractical for system tuning. Accordingly, we have developed the C-TEST toolkit for search evaluation,[1] based on XML testfile and result file formats designed for tuning and lab experiments. These testfiles can formally specify:

- The relative importance of one query to another.
- The relative utility of one result to another.
- The fact that certain groups of documents are near duplicates of each other.
- Different interpretations of the same query.
- The depth of result set which should be compared for this task.

[1]`http://es.csiro.au/C-TEST/`

SIGIR Workshop on the Future of IR Evaluation, July 23, 2009, Boston.

C-TEST testfiles are potentially applicable in many search settings. Here, we focus on the specific problem of generating realistic testfiles for tuning an enterprise search system. Enterprise search is characterised by:

- Well-defined search engine workloads, which we can represent by sampling submitted queries.
- Great diversity, between organisations, in quantity and characteristics of documents to be searched.
- Financial motivation to tune for high performance. Enterprises sometimes spend large sums of money on enterprise search technology in order to boost productivity and competitiveness.

2. PROPOSED METHODS

We propose using a modified n-panel comparison tool (Figure 1). Assuming modest funding, we imagine supplying participants with large ($30''$ if practical), portrait-oriented, high-resolution screens. Care will be needed to position such a screen for usability. This is so that judging depth need not be arbitrarily restricted to ten and that many more results can be displayed without the need for scrolling or page-down actions. The use of two results set from two very different search engines is likely to promote a more thorough enumeration of the set of valuable results.

Logging: As in previous experiments with n-panel evaluation, we would log queries submitted, results clicked and judgments made. The testfile will comprise a sample of logged queries.

Utility tagging: Even with two deep result sets generated by different means, the list of correct answers may not be complete. Because searchers are assumed to be engaged in a real task, they are likely to continue to explore by browsing and further searching. We propose to provide them with a tagging interface in their browser toolbar which will enable them to tag an eventually-found document with the query they consider it to match (selected from a drop-down list based on their recent search history). Since users may not be motivated to tag answers in naturally-occurring searches, we could also use an instrumented browser to record their actions and attempt to detect when an information need is satisfied (e.g. at the end of a session).

Eye gaze tracking: In previous studies, we have looked at results users clicked on, and what features of clicking be-

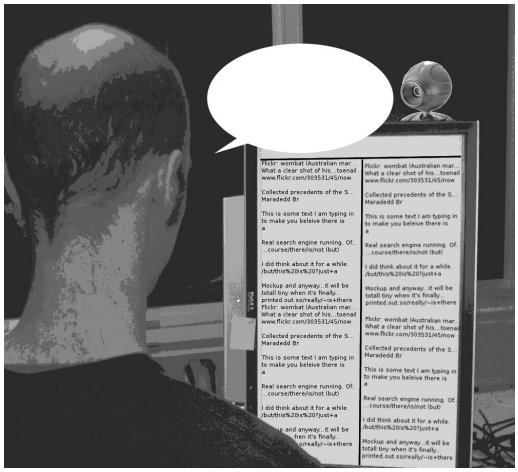

Figure 1: A possible interface for collecting query and judgement information. A two-panel configuration is shown, with logging; gaze tracking and expression recognition via a webcam; and audio feedback via built-in sound equipment.

haviour most accurately predict the explicit judgment actually made. We now propose the use of an eye-gaze tracking facility built into the user's computer to observe which results are actually scanned by the user, detect some measure of attention from pupil diameter, and some indication of degree of cognitive processing from dwell times. Eye gaze reflects attention not selection, and needs to be fused with click data to differentiate between attention-getting bad results and results which are actually useful.

As well as indicating attention, knowing which results are scanned would allow us to choose an appropriate depth.

Audio commentary and feedback: We have previously used pop-up windows to elicit feedback ("You searched for 'IP policy' but so far you haven't clicked on any results. Is that because neither system gave you the answer you wanted?"). In the future we propose using speech generation and recording facilities to ask the user to describe what they are looking for (when they submit a query), and to comment on results they have clicked on. This could be used to enumerate interpretations and to assign utility values to results.

Face expression recognition: Human beings are used to expressing a lot of qualitative information about interactions via facial expressions. It is common to make facial expressions at the screen reflecting some judgements of the information provided, for example the match between expectation and result. The same cameras which are used to detect eye gaze could be used to identify facial expressions and gestures such as nodding or shaking the head.

Labelling and ordering documents: We are developing another approach to assigning utility values to query results. This approach asks subjects assign utility labels to documents, and to then rank them within those labels. Figure 2 shows a prototype interface to support this activity (to be demonstrated at the workshop). Obviously, the n-panel comparison tool would not be used in this activity, but labelling and ordering could be done in-situ and in-context, given cooperative subjects.

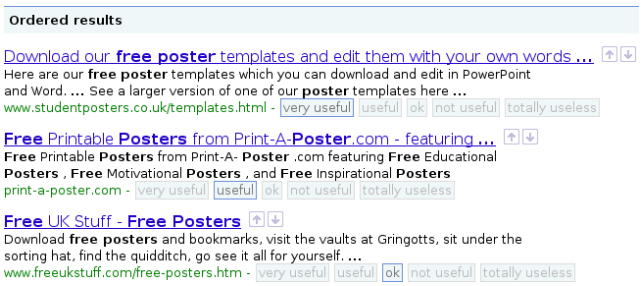

Figure 2: This prototype interface allows the result list to be arranged by usefulness, by a user clicking the up or down button for each result. The subject can also assign labels to results. Label sets can be used to indicate categories of relevance or to identify duplicates or spam.

3. DISCUSSION AND CONCLUSIONS

Any of these methods is of course subject to the bias inherent in selecting subjects. Those with the time and willingness to cooperate may not be representative of the full searching population. Obviously, we will provide the ability for participants to opt-out for particular queries, but this means that particularly important queries (e.g. 'employee retrenchment provisions') are not included.

Enterprise search testfiles are not likely to be made available for general distribution. Knowing what employees of a company are searching for and what documents they have access to, may be valuable competitive intelligence.

Like Cooper [1] we would like to evaluate search systems on the basis of the utility of the answers they provide. If considered appropriate, both "audio commentary and feedback" and "labelling and ordering documents" could be used to elicit utility values in dollars. Our approach replaces Cooper's human experimenter with much cheaper technological alternatives which are on-duty around the clock and arguably less likely to disrupt normal search behaviour.

Unlike Kelly and Belkin [2] our purpose is much narrower and more specific—we want to build testfiles capable of tuning search systems to maximise actual user satisfaction.

Our proposed method extends previous work in n-panel evaluation, by taking advantage of some newly available or newly affordable technology. It has many features in common with studies in a usability lab, but with the vital difference that the experiment is conducted in the workplace, using naturally occurring search needs and in-context judgments. Unlike logfile analysis, our method avoids the need to attempt to interpret or reverse engineer queries submitted and to deduce utility values from uncertain, incomplete, binary-only click data. As a result, we can obtain a representative sample of real workloads and use it to build a more realistic tuning testfile.

4. REFERENCES

[1] W. S. Cooper. On selecting a measure of retrieval effectiveness. *JASIS*, 24(2):87–100, 1973.

[2] D. Kelly and N. J. Belkin. Display time as implicit feedback: Understanding task effects. In *Proc. ACM SIGIR*, pp. 377–384, 2004.

[3] P. Thomas and D. Hawking. Evaluation by comparing result sets in context. In *Proc. CIKM*, pp. 94–101, 2006.