Automated Evaluation of Search Engine Performance via Implicit User Feedback

Himanshu Sharma
Department of Industrial and Manufacturing Engineering
The Pennsylvania State University
University Park, PA, 16801, USA
hus114@psu.edu

Bernard J. Jansen
School of Information Sciences and Technology
The Pennsylvania State University
University Park, PA, 16801, USA
jjansen@ist.psu.edu

ABSTRACT
Measuring the information retrieval effectiveness of Web search engines can be expensive if human relevance judgments are required to evaluate search results. Using implicit user feedback for search engine evaluation provides a cost and time effective manner of addressing this problem. Web search engines can use *human evaluation* of search results without the expense of human evaluators. An additional advantage of this approach is the availability of real time data regarding system performance. We capture user relevance judgments actions such as print, save and bookmark, sending these actions and the corresponding document identifiers to a central server via a client application. We use this implicit feedback to calculate performance metrics, such as precision. We can calculate an overall system performance metric based on a collection of weighted metrics.

Categories and Subject Descriptors
H.3.3 [1] Information Search and Retrieval – relevance feedback

General Terms
Performance, Design, Experimentation, Human Factors

Keywords
Implicit user feedback, search engine evaluation

1. INTRODUCTION
The role of the user is critical in evaluating the retrieval performance of any search engine or information retrieval (IR) system. However, it is not always feasible to obtain relevance judgments from users in environments with universal access and client-server information exchange, such as the Web. Implicit feedback offers a method to gather significant amounts of relevant judgment data for performance evaluation.

The goal of this research is to develop a system that uses implicit user feedback for the real-time evaluation of a search engine's performance. Previous research has focused on the scope and classification of implicit feedback techniques [1] and using implicit feedback as measures of user interest [2]. Oard and Kim [3] identified implicit feedback actions that are indications of user relevance judgments, assuming this data can be captured for evaluation purposes [4].

IR systems, especially high volume traffic Web search engines, can collect large amounts of implicit relevance judgments by recording facets of user interactions between the system and the browser during actual search sessions. Implicit feedback actions can be categorized as (1) *Indicating relevance*, (2) *Indicating non-relevant*, and (3) *Other*. One can classify the retrieved results into four broad categories, namely, (1) *Relevant*, (2) *Not relevant*, (3) *Relevant but already observed*, and (4) *Other*. Using the number of documents categorized by implicit feedback and the number of retrieved results, one can calculate several relevance-based performance metrics for IR systems.

Using a client – server application, we can currently automatically calculate precision-based metrics for a particular search engine. We can extend this application to the calculation of other measures like coverage ratio and comparative recall amongst search engines (i.e. comparing recall between different search engine result listings for the same query), among others measures.

In the following sections, we discuss the application, the current progress, and research results to date. We conclude with future aims and implications of using implicit feedback for IR system evaluation.

2. CURRENT SYSTEM DESCRIPTION
Our research goal is to utilize user search actions as implicit judgments of document relevance. We interpret user interactions with search engine via the browser during the search session as *action-object* pairs. Owing to the popularity and widespread use of Internet Explorer and the Google search engine, our application is customized to capture implicit user judgments for this combination of browser and search engine.

Actions are implicit feedback interactions such as scrolling, saving, printing, bookmarking, adding to favorites, and copying. These interactions occur on *objects* such as documents and passages from documents. An *action-object* pair contains the specific action and an address to the document, which on the Web is a uniform resource locator (URL). We deem certain *actions* such as high scroll count, copy, save, bookmark, and print actions as indications that the *object* is relevant. A wrapper program that interfaces with the browser through Dynamic Data Exchange captures these *action-object* pairs. The wrapper application sends these *action-object* pairs to a server that stores the pairs in a database.

The client-side wrapper program continuously polls through and updates the state of the Internet Explorer window, logging user interactions with the tool bar, changes in the results lists, and changes to the document viewed. Scrolling and dwell time are captured using timers and changes in the cursor position. The wrapper extracts the text of the Internet Explorer status bar to determine whether the user is viewing documents or the search engine results listing. For example, when the user is viewing the result listing, there is no text in the status bar. However, at the moment when the user clicks on a document in the result listing, the status bar text points to the URL of that document. A document clicked upon can be considered as a document viewed by the user. Thus, the application captures the document that the

SIGIR'05, August 15–19, 2005, Salvador, Brazil.
ACM 1-59593-034-5/05/0008.

user is currently viewing in terms of its URL. Also, if the user saves a document, the wrapper captures this action (i.e., *save*) and the corresponding document URL. It then sends this information (i.e., *save URL*) to the server-side application, which in this case would increment the counter for the number of relevant documents. If the user takes no such relevance action after viewing a document, the wrapper records the action of viewing of the documents, but does not record an implicit relevance action on this URL. The server application would annotate this URL as not relevant and increment the counter tracking non-relevant documents.

The server side of the application analyzes the *action-object* pairs, noting which documents are relevant. The application annotates documents already perceived as relevant during the session in a relevant but already observed category. Some actions may indicate non-relevance, such as a user viewing a document, taking no action, but rapidly returning to the search engines. Any URL in the results lists for which the user takes no action, the wrapper application annotates as other (i.e., relevance is undetermined).

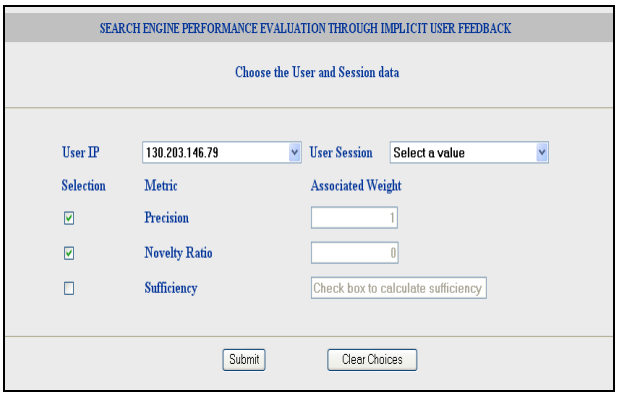

Figure 1. The interface for performance metrics.

Using the server side application, one can assess the performance of the search engine with respect to user relevance judgments through the input page of an active server page (ASP) interface (see Figure 1). This interface allows the selection of specific metrics for the application to calculate, as well the weights assigned to each of these metrics. This permits flexibility for various systems and user bases. For example, precision may be the most important metric for some systems but less important for others. An output page (see Figure 2) allows one to obtain these calculated results, with the option of going back to the input page to change the weights of the selected metrics.

3. SYSTEM EVALUATION

We have conducted an evaluation of the application in an organization setting with 12 manufacturing and quality engineers from a major manufacturing organization. We observed each user conducting distinct searches on the Google search engine with our application running unobtrusively in the background in a naturalistic environment. Initial results have been quite positive and interesting. Presence and absence of a*ction-object* pairs such as *save URL*, *add to favorites URL* and *print URL* indicated relevance and non relevance respectively. The application calculated performance metrics for 18 searches out of the 22 observed searches (efficiency of 81.8%). Metrics were not calculated in the remaining 4 searches due to errors in the capture of user actions such as clicks on a link. We noted that pop up windows opening, incomplete loading of Web pages, slow processor speed and some search behavior by users, sometimes prevent the application from accurately capturing user actions and the URL of documents viewed or clicked upon, leading to erroneously calculated metrics.

The application currently captures many implicit feedback actions; however, "read time" and "scroll" on individual Web pages also appear to be indications of relevance.

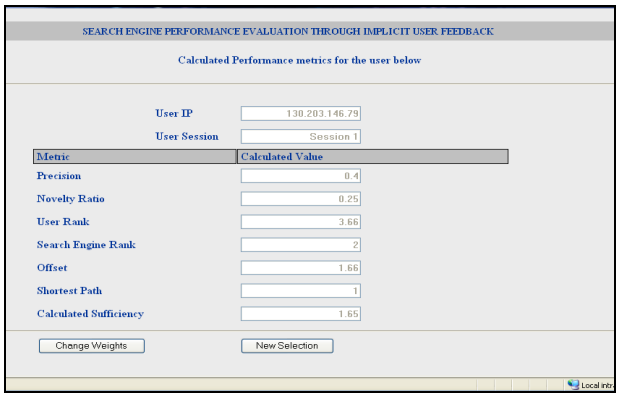

Figure 2. The performance metrics display page.

4. FUTURE DIRECTION OF RESEARCH

The implications and significance of this research is that capturing user interactions as implicit relevance judgments can achieve significant benefits in evaluating search engine performance by reducing the cost and time of evaluating retrieved results.

As in progress research, there are several areas we are currently improving. We are increasing the performance metrics to include comparative recall. More importantly, we are working on processes to automatically capture individual reading and scrolling behaviors. We also aim to enhance the binary relevance judgments with continuous relevance judgments by assigning different weights to various user actions and aggregating among multiple users with the same query.

5. REFERENCES

[1] D. Kelly and J. Teevan, "Implicit Feedback for Inferring User Preference: A Bibliography," *SIGIR Forum*, vol. 37, pp. 18-28, 2003.

[2] S. Dumais, T. Joachims, K. Bharat, and A. Weigend, "SIGIR 2003 Workshop Report: Implicit measures of User Interests and Preferences," *SIGIR Forum*, vol. 37, pp. 50-54, 2003.

[3] D. Oard and J. Kim, "Modeling Information Content Using Observable Behavior," in *Proceedings of the 64th Annual Meeting of the American Society for Information Science and Technology*, Washington, D.C., USA, 2001. pp. 38-45.

[4] R. Villa, M. Chalmers, "A framework for implicitly tracking data", *Proceedings of the Second DELOS Network of Excellence Workshop on Personalisation and Recommender Systems in Digital Libraries, Dublin City University, Ireland, June 2001.* pp 18 – 20.