Proceedings of the

SIGIR 2009 Workshop on Understanding the User – Logging and interpreting user interactions in information search and retrieval

Georg Buscher
DFKI GmbH
georg.buscher@dfki.de

Jacek Gwizdka
Rutgers University
jacekg@rutgers.edu

Jaime Teevan
Microsoft Research
teevan@microsoft.com

Nicholas J. Belkin
Rutgers University
belkin@rutgers.edu

Ralf Bierig
Rutgers University
bierig@rci.rutgers.edu

Ludger van Elst
DFKI GmbH
elst@dfki.uni-kl.de

Joemon Jose
Glasgow University
jj@dcs.gla.ac.uk

1 Introduction

Modern information search systems can benefit greatly from using additional information about the user and the user's behavior, and research in this area is active and growing. Feedback data based on direct interaction (e.g., clicks, scrolling, etc.) as well as on user profiles/preferences has been proven valuable for personalizing the search process, e.g., from how queries are understood to how relevance is assessed. New technology has made it inexpensive and easy to collect more feedback data and more different types of data (e.g., gaze, emotional, or biometric data).

The workshop "Understanding the User – Logging and interpreting user interactions in information search and retrieval" documented in this volume was held in conjunction with the 32nd Annual International ACM SIGIR Conference. It focused on discussing and identifying most promising research directions with respect to logging, interpreting, integrating, and using feedback data. The workshop aimed at bringing together researchers especially from the domains of IR and human-computer interaction interested in the collection, interpretation, and application of user behavior logging for search. Ultimately, one of the main goals was to arrange a commonly shared collection of user interaction logging tools based on a variety of feedback data sources as well as best practices for their usage.

2 Structure of the Workshop

Since one of the main goals of the workshop was to gather practical information and best practices about logging tools, it was structured in a way to foster collaboration and discussion among its participants. Therefore, it was less presentation intensive (it included only 4 oral paper presentations), but contained more collaboration-supporting elements: participant introductions, poster presentations, a panel discussion, and, most importantly, group discussions.

This was also reflected in the types of possible submissions: Experience papers (4 pages) should describe experiences with acquiring, logging, interpreting and/or use of using interaction data. Demos of applications or new technology could be presented. Position statements should focus on types of user interaction data / their interpretation / their use.

Each of those papers and demo descriptions got reviews by two members of the program committee. The program committee also judged the interestingness of each paper with regard to oral presentation (e.g., suitability to spawn discussion). The final selection of the 4 papers for oral presentation was made also with respect to the diversity of topics and approaches they covered. The accepted demos and all remaining accepted papers were selected for poster presentation.

Table 1: Scenarios workshop participants focused on with respect to logging and using (implicit) user interaction data

<table>
<tr>
<td>Types of information interacted with
• Information visualizations / search interfaces
• Web text documents
• Personal information (emails, files on desktop)
• Notes/annotations in documents
• Music
• Images
• Structured or semi-structured data (e.g., medical information)
• Physical content (pictures, books)</td>
<td>Types of (implicit) interaction data
• Queries
• Clicks, URL visits
o Identification of interaction patterns, e.g., repeat actions (repeat queries, repeat URL visits)
• Notes/annotations
• Changes made by author in document
• Eye movements
• Biometric feedback: EEG, galvanic skin response (GSR), facial expressions</td>
</tr>
<tr>
<td colspan="2">Uses of implicit interaction data
• Modeling the user
o Identification of domain knowledge / expertise
o Better expression of interests
o Emotion detection (frustration, stress)
o Identification of good / bad experiences
• Personalization / contextualization
o Improving relevance
o Proactive information delivery
• Introspection / reflection (e.g., analyzing what makes a good searcher)
• Finding better ways to display retrieved information</td>
</tr>
</table>

The program of the workshop also reflected the focus on collaboration: It started with an extended participant introduction session where each participant of the workshop was asked to shortly present his or her main research interests related to the workshop's topics. A poster and demo session followed, succeeded by oral presentations of the 4 selected papers. After each paper, there was limited time for focused questions. In that way, each participant got the chance to see all workshop submissions (either as posters or presentations) and to talk to the authors, after which a panel with 3 panelists was formed based on submitted position statements. Following the panel discussion, breakout groups were formed based on common research interests and practical issues collected

during the participant introduction session. The workshop ended with a summary of the achieved results and next steps to take.

In Table 1, we give an overview of the range of scenarios focused on by the different attendees. Table 2 shows topics the participants were most interested in.

Table 2: Topics of interest

Topics focused on in the above scenarios

- Tools for processing low-level logs (e.g., eye tracking, EEG, ...)
- Ways to combine implicit and explicit feedback data (frameworks)
- Ways (tools) to record context (current task, etc.)
- Sharing of logging tools and log data sets (collection of tools, data formats, etc.)
- Uses for implicit data:
  - Improving information experiences in the aggregate
  - Personalizing information experiences
  - Social sciences: Reflecting on people in the aggregate
  - Introspection: Reflecting on self or individual
- Validity of collected data (collected in the wilds vs. in a user study; dependence on used collection tools)
- Privacy issues

3 Paper, Poster and Demo Presentations

In this section, we group and briefly list the papers that have been accepted for the workshop. Overall, 11 experience papers and 4 demos were accepted which are arranged into 5 topical groups below. Four papers (one from 4 of the 5 groups) were selected for oral presentation.

Logging tools / frameworks

- Oral presentation by Ralf Bierig, Jacek Gwizdka and Michael Cole: *A User-Centered Experiment and Logging Framework for Interactive Information Retrieval*. They presented a framework for multidimensional (interaction) data logging that can be used to conduct interactive IR experiments.
- Demo by Claus-Peter Klas and Matthias Hemmje. *Catching the User - User Context through Live Logging in DAFFODIL*. This demo presented an interactive IR experimentation framework that can be used to log events during a search session such as querying, browsing, storing, and modifying contents on several levels.
- Demo by Robert Capra. *HCI Browser: A Tool for Studying Web Search Behavior*. This demo showed a browser extension that contains the most important functionalities needed when conducting a browser-based user study, such as logging browser-specific events and presenting questionnaires to the user before and after an experiment.
- Demo by Stephen Dignum, Yunhyong Kim, Udo Kruschwitz, Dawei Song, Maria Fasli and Anne De Roeck. *Using Domain Models for Context-Rich User Logging*. The demo presented an interface where users can explore a domain using structured representations thereof. The authors propose using the explored paths of the domain model as contextual feedback.

Analyzing user behavior logs

- Oral Presentation by Robert Capra, Bill Kules, Matt Banta and Tito Sierra. *Faceted Search for Library Catalogs: Developing Grounded Tasks and Analyzing Eye-Tracking Data*. The authors aim at examining how faceted search interfaces are used in a digital library. They conducted an eye tracking user study and discuss challenges and approaches for analyzing gaze data.
- Poster by Hitomi Saito, Hitoshi Terai, Yuka Egusa, Masao Takaku, Makiko Miwa and Noriko Kando. *How Task Types and User Experiences Affect Information-Seeking Behavior on the Web: Using Eye-tracking and Client-side Search Logs*. They used screen-capture logs and eye tracking to identify differences in search behavior according to task type and search experience.
- Poster by Maristella Agosti, Franco Crivellari and Giorgio Maria Di Nunzio. *Evaluation of Digital Library Services Using Complementary Logs*. The authors argue that analyzing query logs alone is not sufficient to study user behavior. Rather, analyzing a larger variety of behavior logs (beyond query logs) and combining them leads to more accurate results.

Analyzing query logs in the aggregate

- Poster by Laura Granka. *Inferring the Public Agenda from Implicit Query Data*. The author presents an approach how to apply query log analysis to create indicators of political interest. As an example, poll ratings of presidential candidates are approximated by query log analysis.
- Poster by Suzan Verberne, Max Hinne, Maarten van der Heijden, Eva D'hondt, Wessel Kraaij and Theo van der Weide. *Annotating URLs with query terms: What factors predict reliable annotations?* The authors try to determine factors that predict the quality of URL annotations from query terms found in query logs.

Interpreting interaction feedback for an improved immediate/aggregated search/browsing experience

- Oral presentation by Mark Cramer, Mike Wertheim and David Hardtke: *Demonstration of Improved Search Result Relevancy Using Real-Time Implicit Relevance Feedback*. The paper reports about Surf Canyon, an existing browser plugin that interprets users' browsing behaviors for immediate improved ranking of results from commercial search engines. They show that incorporating user behavior can drastically improve overall result relevancy in the wild.
- Poster by Rui Li, Evelyn Rozanski and Anne Haake. *Framework of a Real-Time Adaptive Hypermedia System*. The authors present an adaptive hypermedia system that makes use of both browsing behavior and eye movement data of a user while interacting with the system. They use this information to automatically re-arrange information for more suitable user presentation.
- Poster by Max Van Kleek, David Karger and mc Schraefel. *Watching Through the Web: Building Personal Activity and Context-Aware Interfaces using Web Activity Streams*. They use user activity logs from Web-based information to build more personalized activity-sensitive information tools. They particularly focus on activity-based organization of user-created notes.
- Demo by Xuanhui Wang and ChengXiang Zhai. *Massive Implicit Feedback: Organizing Search Logs into Topic Maps for Collaborative Surfing*. In this demo, search and browsing logs from Web searchers are organized into topic maps so that users can follow the footprints from searchers who had similar information needs before.

Behavior-based evaluation measures

- Oral presentation by Emine Yilmaz, Milad Shokouhi, Nick Craswell and Stephen Robertson. *Incorporating user behavior information in IR evaluation*. The authors introduce a new user-centric measure (Expected Browsing Utility, EBU) for information retrieval evaluation which is reconciled with click log information from search engines.
- Poster by Tereza Iofciu, Nick Craswell and Milad Shokouhi. *Evaluating the impact of snippet highlighting in search*. The authors present the idea of highlighting important terms in search

result snippets for helping the user to quickly identify whether a result matches the own query interpretation. They use speed and accuracy of clicks to evaluate the effect of highlighting.

4 Conclusions

Over the course of the workshop, we have seen a great variety of types of logged user interactions, of methods how they are interpreted, and how this information is used and applied. Concerning the latter point, how log data is used and applied, we have seen an especially great variety: from personalization purposes, over a more informed visual design of search systems, to teaching users how to search more effectively.

However, the basis for all those different kinds of applications is the same: logged interaction data between a user and a system. There are basic kinds of interaction data, e.g., based on explicit events from the user while browsing the Web, such as clicks and page transitions as well as mouse movements and scrolling. More advanced and more implicit interaction data logging becomes more and more popular, e.g., based on eye tracking, skin conductance, and EEG. During the workshop, we identified common needs and problems with respect to logging interaction data. They reached from extracting the focused data from different software applications to merging interaction data streams from different sources. Here, we clearly see a need for a common basis of tools and frameworks shared within the community so that individual researchers don't have to re-invent the wheel over and over again.

Acknowledgements

We would like to thank ACM and SIGIR for hosting this workshop as well as the SIGIR workshop committee and especially its chair Diane Kelly for their very helpful feedback. We are further very thankful to the authors, the members of our program committee, and all participants. They helped to form a very lively, spirited, highly interesting, and successful workshop.

Program Committee

- Eugene Agichtein (Emory University, Canada)
- Richard Atterer (University of Munich, Germany)
- Nick Craswell (Microsoft Research, England)
- Susan Dumais (Microsoft Research, USA)
- Laura Granka (Stanford, Google Inc., USA)
- Kirstie Hawkey (UBC, Canada)
- Eelco Herder (L3S, Germany)
- Thorsten Joachims (Cornell University, USA)
- Melanie Kellar (Google Inc., USA)
- Douglas Oard (University of Maryland, USA)

Demonstration of Improved Search Result Relevancy Using Real-Time Implicit Relevance Feedback

David Hardtke
Surf Canyon Incorporated
274 14th St.
Oakland, CA 94612
hardtke@surfcanyon.com

Mike Wertheim
Surf Canyon Incorporated
274 14th St.
Oakland, CA 94612
mikew@surfcanyon.com

Mark Cramer
Surf Canyon Incorporated
274 14th St.
Oakland, CA 94612
mcramer@surfcanyon.com

ABSTRACT

Surf Canyon has developed real-time implicit personalization technology for web search and implemented the technology in a browser extension that can dynamically modify search engine results pages (Google, Yahoo!, and Live Search). A combination of explicit (queries, reformulations) and implicit (clickthroughs, skips, page reads, etc.) user signals are used to construct a model of instantaneous user intent. This user intent model is combined with the initial search result rankings in order to present recommended search results to the user as well as to reorder subsequent search engine results pages after the initial page. This paper will use data from the first three months of Surf Canyon usage to show that a user intent model built from implicit user signals can dramatically improve the relevancy of search results.

Keywords

Implicit Relevance Feedback, Personalization, Adaptive Search System

1. INTRODUCTION

It has long since been demonstrated that *explicit* relevance feedback can improve both precision and recall in information retrieval[1]. An initial query is used to retrieve a set of documents. The user is then asked to manually rate a subset of the documents as relevant or not relevant. The terms appearing in the relevant document are then added to the initial query to produce a new query. Additionally, non-relevant documents can be used to remove or de-emphasize terms for the reformulated query. This process can be repeated iteratively, but it was found that after a few iterations very few new relevant documents are found [2].

Explicit relevance feedback as described above requires active user participation. An alternative method that does not require specific user participation is *pseudo* relevance feedback. In this scheme, the top N documents from the initial query are assumed to be relevant. The important terms in these documents are then used to expand the original query.

Implicit Relevance Feedback aims to improve the precision and recall of information retrieval by utilizing user actions to infer the relevance or non-relevance of documents. Many different user behavior signals can contribute to a probabilistic evaluation of document relevance. Explicit document relevance determinations are more accurate, but implicit relevance determinations are more easily obtained as they require no additional user effort.

SIGIR '09, July 19-23, 2009, Boston, USA.

2. IMPLICIT SIGNALS AND USER INFORMATION NEED

With the large, open nature of the World Wide Web it is very difficult to evaluate the quality of search engine algorithms using explicit human evaluators. Hence, there have been numerous investigations into using implicit user signals for evaluation and optimization of search engine quality. Several studies have investigated the extent to which a clickthrough on a specific search engine result can be interpreted as a user indication of document relevancy (for a review see [3]). The primary issue involving clickthrough data is that users are most likely to click on higher ranked documents because they tend to read the SERP (search engine results page) from top to bottom. Additionally, users trust that a search engine places the most relevant documents at the highest positions on the SERP.

Joachims *et al* used eye tracking studies combined with manual relevance judgements to investigate the accuracy of clickthrough data for implicit relevance feedback [4]. They conclude that clickthrough data can be used to accurately determine relative document relevancies. If, for instance, a user clicks on a search result after skipping other search results, subsequent evaluation by human judges show that in $\sim$80% of cases the clicked document is more relevant to the query than the documents that were skipped.

In addition to clickthroughs, other user behaviors can be related to document relevancy. Fox *et al.* used a browser add-in to track user behavior for a volunteer sample of office workers[5]. In addition to tracking their search and web usage, the browser add-in would prompt the user for specific relevance evaluations for pages they had visited. Using the observed user behavior and subsequent relevance evaluations, they were able to correlate implicit user signals with explicit user evaluations and determine what user signals are most likely to indicate document relevance. For pages clicked by the user, the user indicated that they were either satisfied or partially satisfied with the document nearly 70% of the time. In the study, two other variables were found to be most important for predicting user satisfaction with a result page visit. The first was the duration of time that

the user spent away from the SERP before returning – if the user was away from the SERP for a short period of time they tended to be dissatisfied with the document. The other important variable for predicting user satisfaction was the "Exit type" – users that closed the browser on a result page tended to be satisfied with that result page. The important outcome of this and other studies is that implicit user behavior can be used instead of explicit user feedback to determine the user's information need.

3. IMPLICIT REAL-TIME PERSONALIZATION

As discussed in the previous section, it has been shown that implicit user behavior can often infer satisfaction with visited results pages. The goal of the Surf Canyon technology is to use implicit user behavior to predict which *unseen* documents in a collection are most relevant to the user and to recommend these documents to the user.

Shen, Tan, and Zhai[1] have investigated context-sensitive adaptive information retrieval systems [6]. They use both clickthrough information and query history information to update the retrieval and ranking algorithm. A TREC collection was used since manual relevancy judgements are available. They built an adaptive search interface to this collection, and had 3 volunteers conduct searches on 30 relatively difficult TREC topics. The users could query, re-query, examine document summaries, and examine documents. To quantify the retrieval algorithms, they used Mean Average Precision (MAP) or Precision at 20 documents. As these were difficult TREC topics, users submitted multiple queries for each topic. They found that including query history produced a marginal improvement in MAP, while use of clickthrough information produced dramatic increases (up to nearly 100%) in MAP.

Shen *et al.* also built an experimental adaptive search interface called UCAIR (User-Centered Adaptive Information Retrieval) [7]. Their client-side search agent has the capability of automatic query reformulation and active reranking of unseen search results based on a context driven user model. They evaluated their system by asking 6 graduate students to work on TREC topic distillation tasks. At the end of each topic, the volunteers were asked to manually evaluate the relevance of 30 top ranked search results displayed by the system. The top results shown are mixed between Google rankings and UCAIR rankings (some results overlap), and the evaluators could not distinguish the two. UCAIR rankings show a 20% increase in precision for the top 20 results.

The Surf Canyon browser extension represents the first attempt to integrate implicit relevance feedback directly into the major commercial search engines. Hence, we are able to evaluate this technology *outside* of controlled studies. From a research perspective, this is the first study to investigate this technology in the context of normal searches by normal users. The drawback is that we have no chance to collect *a posteori* relevancy judgements from the searchers or to conduct surveys to evaluate the user experience. We can, however, quickly collect large amounts of user data in order to evaluate the technology.

[1]Shen, Tan, and Zhai are co-authors on one Surf Canyon patent application but were not actively involved in the work presented here

4. TECHNOLOGICAL DETAILS

Surf Canyon's technology can be used as both a traditional web search engine and as a browser extension that dynamically modifies the search results page from commercial search engines (currently Google, Yahoo!, and Live Search). The underlying algorithms in the two cases are mostly identical. As the data presented was gathered using the browser extension, we will describe that here.

Surf Canyon's browser extension was publicly launched on February 19, 2008. From that point forward visitors to the Surf Canyon website[2] were invited to download a small piece of free software that is installed in their browser. The software works with both Internet Explorer and Firefox. Although the implementation differs for the two browsers, the functionality is identical.

Internet Explorer leads in all current studies of web browser market share with March 2008 market share estimated between 60% and 90%. Among users of the Surf Canyon browser extension, however, about 75% use Firefox. Among users who merely visit the extension download page, the breakdown by browser type is nearly 50/50. Part of the skew towards Firefox in both website visitors and users of the product can be attributed to the fact that marketing of the product has been mainly via technology blogs. Readers of technology blogs are more likely to use operating systems for which Internet Explorer is not available (e.g. Mac, Linux). Additionally, we speculate that Firefox may be more prevalent among readers of technology blogs. The difference between the fraction of visitors to the site using Firefox (∼50%) and the fraction of people who install and use the product using Firefox (∼75%) is likely due to the more widespread acceptance towards browser extensions in the Firefox community. The Firefox browser was specifically designed to have minimal core functionality augmented by browser addons submitted by the developer community. The technologies used to implement Internet Explorer browser extensions are also often used to distribute malware so there may be a higher level of distrust among IE users.

Once the browser extension is installed, the user never needs to visit the company web site again to use the product. The user enters a Google, Yahoo!, or Live Search web search query just as they would for any search (using either the search bar built into the browser or by navigating to the URL of the search engine). After the initial query, the search engine results page is returned exactly as it would be were Surf Canyon not installed (for most users who have not specified otherwise, the default number of search results is 10). Two minor modifications are made to the SERP. Small bull's eyes are placed next to the title hyperlink for each search result (see Figure 1). Also, the numbered links to subsequent search engine results pages at the bottom of the SERP are replaced by a single "More Results" link.

The client side browser extension is used to communicate with the central Surf Canyon servers and to dynamically update the search engine results page. The personalization algorithms currently reside on the Surf Canyon servers. This client-server architecture is used primarily to facilitate optimization of the algorithm and to support active research studies. Since web search patterns vary widely by user, the best way to evaluate personalized search algorithms is to vary the algorithms on the same set of users while main-

[2]http://www.surfcanyon.com

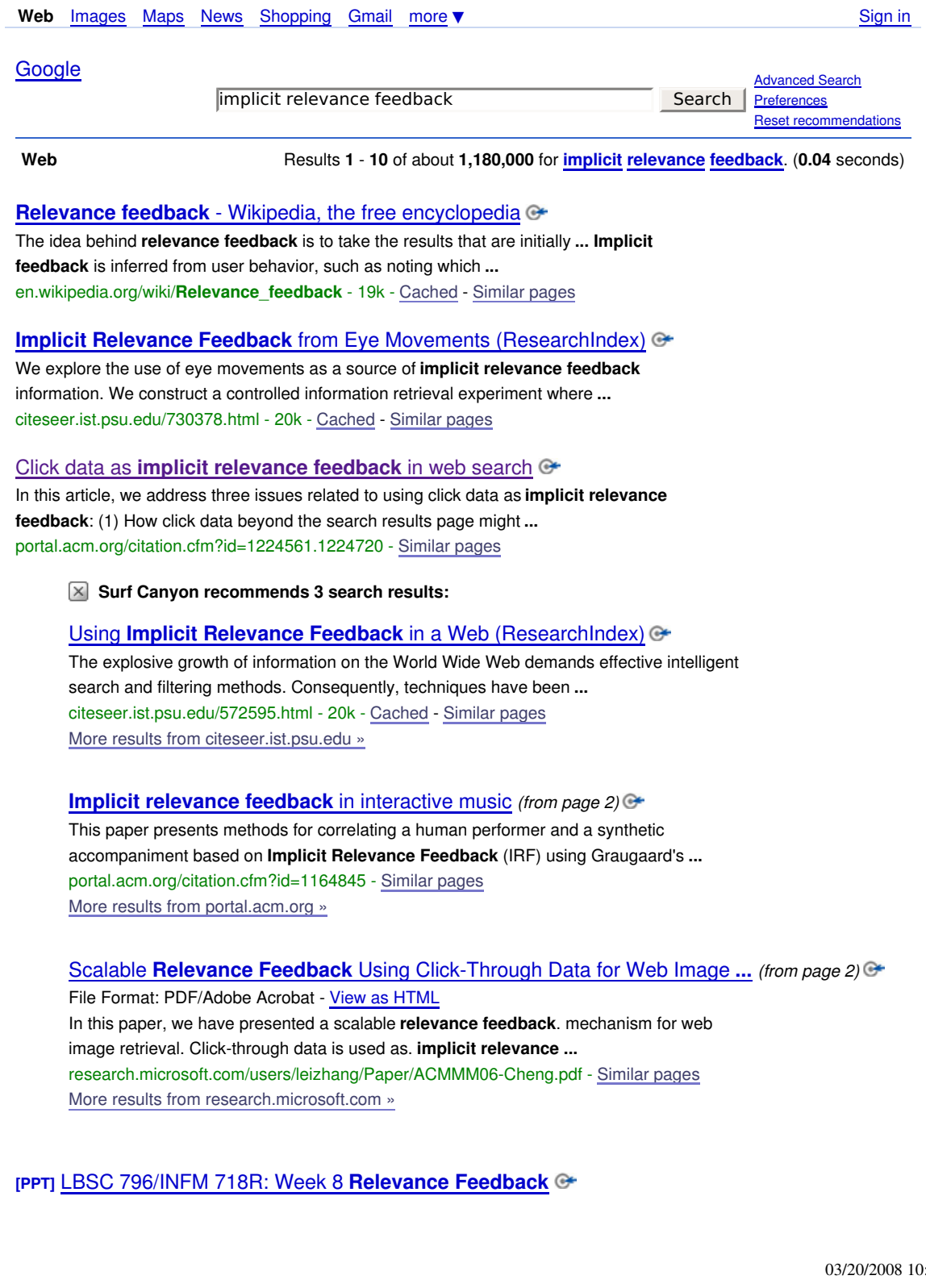


Figure 1: A screenshot of the Google search result page with Surf Canyon installed. The third link was selected by the user, leading to three recommended search results.

taining an identical user interface. With the client-server architecture, the implicit relevance feedback algorithms can be modified without alerting the user to any changes. Nothing fundamental prevents the technology from becoming exclusively client side.

In addition to the ten results displayed by the search engine to the user, a larger set of results (typically 200) for the same query is gathered by the server. With few exceptions, the top 10 links in the larger result set are identical to the results displayed by the search engine. While the user reads the search result page, the back-end servers parse the larger result set and prepare to respond to user actions. Each user action on the search result page is sent to the back-end server (note that we are only using the user's actions on the SERP for personalization and do not follow the user after they leave the SERP). For certain actions (select a link, select a Surf Canyon bull's eye, ask for more results) the back end server sends recommended search results to the browser. The Surf Canyon real-time implicit personalization algorithm incorporates both the initial rank of the result and personalized instantaneous relevancies. The implicit feedback signals used to calculate the real-time search result ranks are cumulative across all recent related queries by that user. The algorithm does not, however, utilize any long-term user profiling or collaborative filtering. The precise details of the Surf Canyon algorithm are proprietary and are not important for the evaluation of the technology presented below. If an undisplayed result from the larger set of results is deemed by Surf Canyon's algorithm to be more relevant than other results displayed below the last selected link, it is shown as an indented recommendation below the last selected link.

The resulting page is shown in Figure 1. Here, the user entered a query for "implicit relevance feedback" on Google[3]. Google returned 10 organic search results (only three of which are displayed in Figure 1) of the 1,180,000 documents in their web index that satisfy the query. The user then selected the third organic search result, a paper from an ACM conference entitled "Click data as implicit relevance feedback in web search". Based on the implicit user signals (which include interactions with this SERP, recent similar queries, and interactions with those results pages) the Surf Canyon algorithm recommends three search results. These links were initially given a higher initial rank (> 10) by the Google algorithm in response to the query "implicit relevance feedback". The real-time personalization algorithm has determined, however, that the three recommended links are more pertinent to this user's information need at this particular time than the results displayed by Google with initial ranks 4-10.

Recommendations are also generated when a user clicks on the small bull's eyes next to the link title. We assume that a selection of a bull's eye indicates that the linked document is similar to but not precisely what the user is looking for. For the analysis below, up to three recommendations are generated for each link selection or bull's eye selection. Unless the user specifically removes recommended search results by clicking on the bull's eye or by clicking the close box, they remain displayed on the page. Recommendations can nest up to three levels deep – if the user clicks on the first recommended result then up to three recommendations are generated immediately below this search result.

At the bottom of the 10 organic search results, there is a link to get "More Results". If the user requests the next page of results, all results shown on the second and subsequent pages are determined using Surf Canyon's instantaneous relevancy algorithm. Unlike the default search engine behavior, subsequent pages of results are added to the existing page. After selecting "More Results" links 1-20 are displayed in the browser, with link 11 focused at the top of the window (the user needs to scroll up to see links 1-10).

5. ANALYSIS OF USER BEHAVIOR

Most previous studies of Interactive Information Retrieval systems have used post-search user surveys to evaluate the efficacy of the systems. These studies also tended to recruit test subjects and use closed collections and/or specific research topics. The data presented here was collected from an anonymous (but not necessarily representative) set of web surfers during the course of their interactions with the three leading search engines (Google, Yahoo, and Live Search). The majority of searches were conducted using Google. Where possible, we have analyzed the user data independently for each of the search engines and have not found any cases where the conclusions drawn from this study would differ depending on the user's choice of search engine. The total number of unique search queries analyzed was $\sim$700,000.

Since the users in this study were acquired primarily from technology web blogs, their search behavior can be expected to be significantly different than the average web surfer. Thus, we cannot evaluate the real-time personalization technology by comparing to previous studies of web user behavior. Also, since we have changed the appearance of the SERP and also dynamically modify the SERP, any metrics calculated from our data cannot be directly compared to historical data due to the different user interface.

Surf Canyon only shows recommendations after a bull's eye or search result is selected. It is therefore interesting to investigate how many actions a user makes for a given query as this tells us how frequently implicit personalization within the same query can be of benefit. Jansen and Spink [8] found from a meta-analysis of search engine log studies that user interaction with the search engine results pages is decreasing. In 1997, 71% of searchers viewed beyond the first page of search results. In 2002 only 27% of searchers looked past the first page of search results. There is a paucity of data on the number of web pages visited per search. Jansen and Spink [9] reported the mean number of web pages visited per query to be 2.5 for AllTheWeb searches in 2001, but they exclude queries where no pages were visited in this estimate. Analysis of the AOL query logs from 2006 [10] gives a mean number of web pages viewed per unique query of 0.97. For the current data sample, the mean number of search results visited is 0.56. The comparatively low number of search results that were selected in the current study has multiple partial explanations. The search results page now contains multiple additional links (news, videos) that are not counted in this study. Additionally, the information that the user is looking for is often on the SERP (e.g. a search for a restaurant often produces the map, phone number, and address). Search engines have replaced bookmarks and direct URL typing for re-visiting web sites. For such navigational searches the user will have either one or zero

[3]http://www.google.com

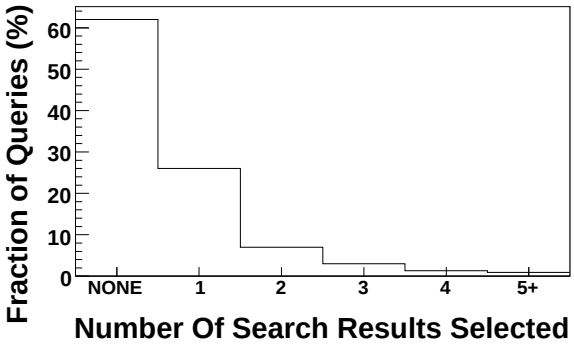


Figure 2: Distribution of total number of selections per query.

clicks depending on whether the specific web page is listed on the SERP. Additionally, it may be that the current sample of users is biased towards searchers who are less likely to click on links.

Figure 2 shows the distribution of the total number of selections per query. 62% of all queries lead to the selection of zero search results. Since Surf Canyon does nothing until after the first selection, this number is intrinsic to the current users interacting with these particular search engines. A recent study by Downey, Dumais and Horvitz also showed that after a query the user's next action is to re-query or end the search session about half the time [11]. In our study, only 12% of queries lead to more than one user selection. A goal of implicit real-time personalization would be to decrease direct query reformulation and to increase the number of informational queries that lead to multiple selections. The current data sample is insufficient to study whether this goal has been achieved.

In order to evaluate the implicit personalization technology developed by Surf Canyon we chose to compare the actions of the same set of users with and without the implicit personalization technology enabled. Our baseline control sample was created by randomly replacing recommended search results with random search results selected from among the results with initial ranks 11-200. These "Random Recommendations" were only shown for 5% of the cases where recommendations were generated. The position (1, 2, or 3) in the recommendation list was also random. These random recommendations were not necessarily poor, as they do come from the list of results generated by the search engine in response to the query.

Figure 3 shows the click frequency for Surf Canyon recommendations as a function of the position of the recommendation relative to the last selected search result. Position 1 is immediately below the last selected search result. Also shown are the click frequencies for "Random Recommendations" placed at the same positions. In both cases, the frequency is relative to the total number of recommendations shown at that position. The increase in click rate (∼60%) is constant within statistical uncertainties for all recommended link positions. Note that the recommendations are generated each time a user selects a link and are considered to be shown even if the user does not return to the SERP. The low absolute click rates (3% or less) are due to the fact that users do not often click on more than one search result as discussed above. The important point, however, is that the Surf Canyon implicit relevance feedback technology increases the click frequency by ∼80% compared to the links presented without any real-time user-intent modelling. The relative increase in clickthrough rate is constant (within statistical errors) for all display positions even though the absolute clickthrough rates rapidly drop as funciton of display position.

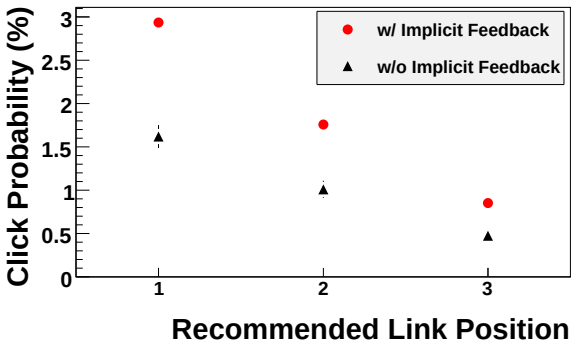


Figure 3: Probability (%) that a recommended search result will be clicked as a function of display position relative to the last selected search result. The red circles are for recommendations selected using Surf Canyon's instantaneous relevancy algorithm, while the black triangles are for the random control sample that does not incorporate relevance feedback.

Figure 4 shows the per query distribution of initial search result ranks for all selected search links in the current data sample. The top 10 links are selected most frequently. Search results beyond 10 are all displayed using Surf Canyon's algorithm (either through a bull's eye selection, a link selection, or when the user selects more results). For the results displayed by Surf Canyon (initial ranks > 10), the selection frequency follows a power-law distribution with $P(IR) = 38\% * IR^{-1.8}$, where IR is the initial rank.

As Surf Canyon's algorithm favors links with higher initial rank, the click frequency distribution does not fully reflect the relevancy of the links as a function of initial rank. Figure 5 shows the probability that a shown recommendation is clicked as a function of the initial rank. This is only for recommendations shown in the first position below the last selected link. After using Surf Canyon's instantaneous relevancy algorithm, this probability shows at most a weak dependence on the initial rank of the search result. The dotted link shows the result of a linear regression to the data, $P(IR) = 3.2 - (0.0025 \pm 0.00101) * IR$. When sufficient data is available we will repeat the same analysis for "Random Recommendations" as that will give us a user-interface independent estimate of the relative relevance for deep links in the search result set before the application of the implicit feedback algorithms.

For the second and subsequent results pages, the browser extension has complete control over all displayed search results. For a short period of time we produced search results pages that mixed Surf Canyon's top ranked results with results having the top initial ranks from the search

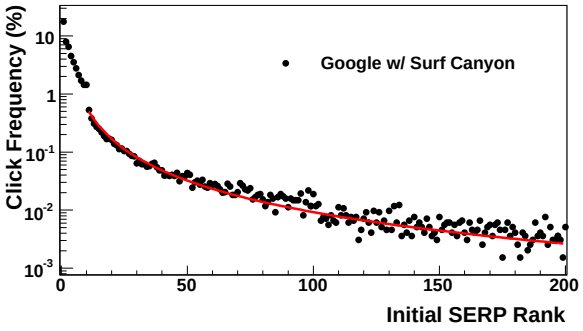


Figure 4: Frequency per non-repeated search query for link selection as a function of initial search result rank.

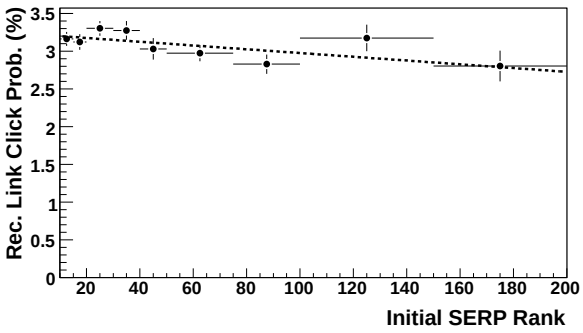


Figure 5: Probability that a *displayed* recommended link is selected as a function of the initial search result rank. This data only include links from the first position immediately below the last selected search result.

engine. This procedure was proposed by Joachims as a way to use clickthrough data to determine relative user preference between two search engine retrieval algorithms [12]. Each time a user requests "More Results", two lists are generated. The first list (SC) contains the remaining search results as ranked by the Surf Canyon's instantaneous relevancy algorithm. The second list (IR) contains the same set of results ranked by their initial display rank from the search engine. The list of results shown to the user is such that the top k_{SC} and k_{IR} results are displayed from each list, with $|k_{SC} - k_{IR}| < 1$. Whenever $k_{SC} = k_{IR}$ the next search result is taken from one of the lists chosen at random. Thus, the topmost search result on the second page will reflect Surf Canyon's ranking half the time and the initial search result order half the time. By mixing the search results this way, the user will see, on average, an equal number of search results from each ranking algorithm in each position on the page. The users have no way of determining which algorithm produced each search result. If the users select more search results from one ranking algorithm compared to the other ranking algorithm it demonstrates an absolute user preference for the retrieval function that led to more selections.

Figure 6 shows the ratio of link clicks for the two retrieval functions. IR is the retrieval function based on the result rank returned from the search engine. SC is the retrieval function incorporating Surf Canyon's implicit relevance feedback technology. The ratio is plotted as a function of the number of links selected previously for that query. Previously selected links are generally considered to be positive content feedback. If, on the other had, no links were selected then the algorithm bases its decision exclusively on negative feedback indications (skipped links) and on the user intent model that may have been developed for similar recent related queries.

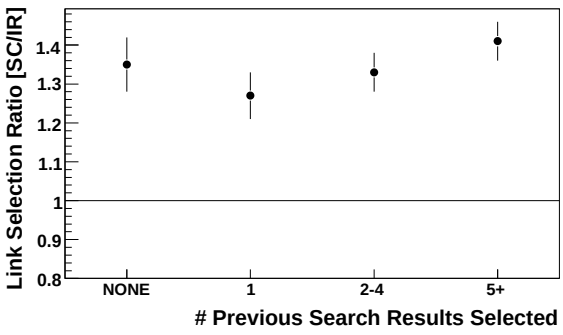


Figure 6: Ratio of click frequency for second and subsequent search results page links ordered by Surf Canyon's Implicit Relevance Feedback algorithm (SC) compared to links ordered by the initial search engine result rank (IR).

We observe that, independent of the number of previous user link selections in the same query, the number of clicks on links from the relevance feedback algorithm is higher than links displayed because of their higher initial rank. This demonstrates an absolute user preference for the ranking algorithm that utilizes implicit relevance feedback. Remark-

ably, the significant user preference for search results retrieved using the implicit feedback algorithm is also apparent when the user had zero positive clickthrough actions on the first 10 results. After skipping the first 10 results and asking for a subsequent set of search links, the users are ∼35% more likely to click on the top ranked Surf Canyon result compared to result # 11 from Google. Clearly, the searcher is not so interested in search results produced by the identical algorithm that produced the 10 skipped links and an update of the user intent model for this query is appropriate.

6. CONCLUSIONS AND FUTURE DIRECTIONS

Surf Canyon is an interactive information retrieval system that dynamically modifies the SERP from major search engines based on implicit relevance feedback. This was built with the goal of relieving the growing user frustration with the search experience and to help searchers "find what they need right now". The system presents recommended search results based on an instantaneous user-intent model. By comparing clickthrough rates, it was shown that real-time implicit personalization can dramatically increase the relevancy of presented search results.

Users of web search engines learn to think like the search engines they are using. As an example, searchers tend to select words with high IDF (inverse document frequency) when formulating queries – they naturally select the rarest terms that they can think of that would be in all documents they desire. Excellent searchers can often formulate sufficiently specific queries after multiple iterations such that they eventually find what they need. Properly implemented implicit relevance feedback would reduce the need for query reformulations, but it should be noted that in the current study most users had not yet adjusted their browsing habits to the modified behavior of the search engine. By tracking the current users in the future we hope to see changes in user behavior that can further improve the utility of this technology. As the user-intent model is cumulative, more interaction will produce better recommendations *once* the users learn to trust the system.

7. REFERENCES

[1] J.J. Rocchio. *The Smart Retrieval System Experiments in Automatic Document Processing.* Prentice Hall, 1971.

[2] D. Harman. Relevance feedback revisited. In *Proceedings of the Fifteenth International ACM SIGIR Conference*, pages 1–10, 1992.

[3] D. Kelly and J. Teevan. Implicit feedback for inferring user preference: A bibliography. In *SIGIR Forum 37(2)*, pages 18–28, 2003.

[4] Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. Accurately interpreting clickthrough data as implicit feedback. In *SIGIR '05*, 2005.

[5] Steve Fox, Kuldeep Karnawat, Mark Mydland, Susan Dumais, and Thomas White. Evaluating implicit measures to improve web search. *ACM Transactions on Information Systems*, 23(2):147–168, April 2005.

[6] Xuehua Shen, Bin Tan, and ChengXiang Zhai. Context-sensitive information retrieval using implicit feedback. In *SIGIR '05*, 2005.

[7] Xuehua Shen, Bin Tan, and ChengXiang Zhai. Implicit user modelling for personalized search. In *CIKM '05*, 2005.

[8] B. Jansen and A. Spink. How are we searching the world wide web?: a comparison of nine search engine transaction logs. *Information Processing and Management*, 42(1):248–263, 2006.

[9] B. Jansen and A. Spink. An analysis of web documents retrieved and viewed. In *The 4th International Conference on Internet Computing*, pages 65–69, 2003.

[10] G. Pass, A. Chowdhury, and C. Torgeson. A picture of search. In *The First International Conference on Scalable Information Systems*, 2006.

[11] D. Downey, S. Dumais, and E. Horvitz. Studies of web search with common and rare queries. In *SIGIR '07*, 2007.

[12] T. Joachims. Unbiased evaluation of retrieval quality using clickthrough data. In *SIGIR Workshop on Mathematical/Formal Methods in Information Retrieval*, 2002.

A User-Centered Experiment and Logging Framework for Interactive Information Retrieval [*] [†]

Ralf Bierig
SC&I Rutgers University
4 Huntington St.,
New Brunswick
NJ 08901, USA
bierig@rci.rutgers.edu

Jacek Gwizdka
SC&I Rutgers University
4 Huntington St.,
New Brunswick
NJ 08901, USA
jgwizdka@scils.rutgers.edu

Michael Cole
SC&I Rutgers University
4 Huntington St.,
New Brunswick
NJ 08901, USA
mcole@scils.rutgers.edu

ABSTRACT

This paper describes an experiment system framework that enables researchers to design and conduct task-based experiments for Interactive Information Retrieval (IIR). The primary focus is on multidimensional logging to obtain rich behavioral data from participants. We summarize initial experiences and highlight the benefits of multidimensional data logging within the system framework.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

Keywords

User logging, Interactive Information Retrieval, Evaluation

1. INTRODUCTION

Over the last two decades, Interactive Information Retrieval (IIR) has established a new direction within the tradition of IR. Evaluation in traditional IR is often performed in laboratory settings where controlled collections and queries are evaluated against static information needs. IIR introduces the user at the center of a more naturalistic search environment. Belkin and colleagues [3, 2] suggested the concept of an information seeking episode composed of a sequence of a person's interactions with information objects, determined by a specific goal, conditioned by an initial task, the general context and the more specific situation in which the episode takes place, and the application of a particular information seeking strategy.

This poses new challenges for the evaluation of information retrieval systems. An enriched set of possible user behaviors needs to be addressed and included as part of the evaluation process. Systems need to address information about the entire interactive process with which users' accomplish a task. This problem has so far only been initially explored [4].

This paper describes an experiment system framework that enables researchers to design and conduct task-based IIR experiments. The paper is focused on the logging features of the system designed to obtain rich behavioral data from participants. The following section describes the overall architecture of the system. Section 3 provides more details about its specific logging features. Section 4 summarizes initial experiences with multidimensional data logging within the system framework based on initial data analysis from three user studies. Future work is proposed in section 5.

2. THE POODLE IIR EXPERIMENT SYSTEM FRAMEWORK

The PooDLE IIR Experiment System Framework is part of an the ongoing research project. The goal of PooDLE[1] to investigate ways to improve information seeking in digital libraries; the analysis concentrates on an array of interacting factors involved in such online search activities. The overall aim of the framework is to reduce the complexity of designing and conducting IIR experiments using multidimensional logging of users' interactive search behavior. Such experiments usually require a complex arrangement of system components (e.g. GUI, user management and persistent data storage) including logging facilities that monitor implicit user behavior. Our framework enables researchers to focus on the design of the experiment including questionnaire and task design and the selection of appropriate logging tools. This can help to reduce the overall time and effort that is needed to design and conduct experiments that support the needs for IIR. As shown in figure 1, the experiment system framework consists of two sides – a server that operates in an Apache webserver environment and a client that resides on the machine where the experiment is conducted. We distinguish the following components:

- *Login and Authentication* manages participants, allows them to authenticate with the system, and enables the system to direct individuals to particular experiment

SIGIR'09, July 19-23, 2009,Boston, USA.

[†]This work is supported, in part, by the Institute of Museum and Library Services (IMLS grant LG-06-07-0105-07)

[1]http://www.scils.rutgers.edu/imls/poodle/index.html

.

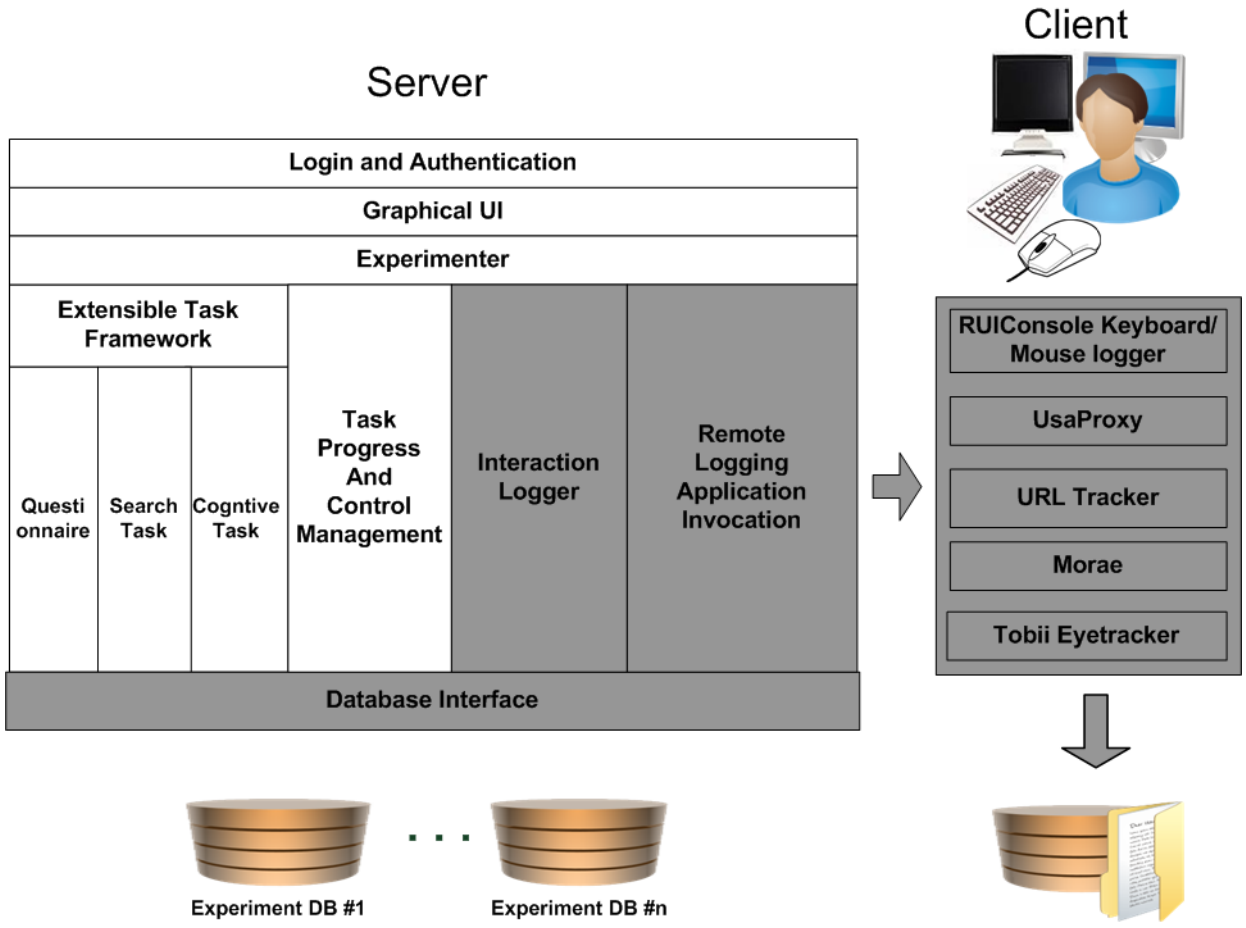


Figure 1: System components of the PooDLE IIR Experiment System Framework. Logging features highlighted in grey.

setups; multiple experiments may exist and users can be registered for multiple or multi-part experiments at any time.

- The *Graphical UI* allows participants to authenticate with the framework and activate their experiment. Each experiment consists of a number of rotated tasks that are provided with a generic menu that presents the predefined task order to the user. After every completed task, the UI guides the participant back to the menu that now highlights the completed tasks. This allows participants to navigate between tasks and gain feedback that helps them to track their progress. In addition, the interface presents participants with additional information, instructions and warnings when progressing through the tasks of an experiment.
- The *Experimenter* controls and coordinates the core components of the system – these are:
  - An *Extensible Task Framework* that provides a range of standard tasks for IIR experiments that are part of the framework (e.g. questionnaires for acquiring background information and general feedback from participants, search tasks with a bookmarking feature and an evaluation procedure, and cognitive tasks to obtain information about individual differences between participants). Tasks are easily added to this basic collection and can be reused as part of the framework in different experiments.
  - The *Task Progress and Control Management* provides participants with (rotated) task sequences, monitors their state within the experiment, and allows them to continue interrupted experiments at a later point in time.
  - The *Interaction Logger* allows tasks to register and trigger logging messages at strategic points within the task. The system automatically logs the beginning and end of each task at task boundaries.
  - *Remote Logging Application Invocation* calls logging applications that reside on the client. This allows for rich client-sided logging of low level user behavior obtained from specific hardware (e.g. mouse movements or eye-tracking information).
- The Database interface manages all access to one or more databases that store users' interaction logs as

well as the basic experiment design for other system components (e.g. participants, tasks and experiment blocks in the form of task rotations for individual users).

3. USER INTERACTION LOGGING

This section focuses on the logging features of the Experiment System Framework as highlighted in grey in figure 1. The logging features and the arrangement of logging tools within the framework have been informed by the following requirements:

- *Hybridity:* All logging functionality is divided between a more general server architecture and a more specific client; this integrates server-based as well as client-based logging features into a hybrid system framework. Whereas the server logs user interactions uniformly across experiments, client logging is targeted to the capabilities of the particular client machine used for the experiment. Researchers can select from a range of logging tools or integrate their own tools to record user behavior. This enables the system to use low level input devices, normally inaccessible by the server, to be controlled by logging tools residing on the client.

- *Flexibility:* Client logging tools can be combined through a loosely coupled XML-based configuration that is provided at task granularity. The system framework uses these task configurations to start logging tools on the client when the participant enters a task and stops them when the participant completes a task. This gives researchers the flexibility to compose logging tools as part of the experiment design and attach them to the configuration of the task. Such configurations can later be reused as design templates which promotes uniformly across experiments and ensures important types of user interaction data are being logged.

- *Scalability:* Experiments can be configured to apply a number of different client machines as part of the data collection. A researcher can, for example, trigger another client computer to record video from a second web camera or simultaneously activate several clients for experiments that involve more than one participant. Redundant instances of the same logging tools can be instantiated to produce multiple data streams to overcome potential measurement errors and instabilities on a data stream due to load or general failure of hard and software.

The client is configured to work with the following selection of open-source and commercial logging tools that record different behavioral aspects of participants:

- *RUIConsole* is an adapted command line version of the RUI tools developed at Pennsylvania State University [5]. RUI logs low level mouse movements, mouse clicks, and keystrokes. Our extension additionally provides full control over its logging features through a command line interface to allow for more efficient automated use within our experiment framework.

- *UsaProxy* is a javascript based HTTP proxy developed at the University of Munich [1] that logs interactive user behavior unobtrusively through injected javascript. It monitors page loads as well as resize and focus events. It identifies mouse hover events over page elements, mouse movements, mouse clicks, keystrokes, and scrolling. Our version of UsaProxy is slightly modified as we don't log mouse movements with this tool. UsaProxy can run directly on the client, but can also be activated on a separate computer to balance load.

- The *URL Tracker* is a command line tool that extracts and logs the users current web location directly from the Internet Explorer (IE) address bar and makes it available to the system framework. This allows any task to determine participants' current position on the web and to monitor their browsing history within a task.

- *Tobii Eyetracker:* We use the Tobii T60 eyetracking hardware which is packaged with Tobii Studio[2], a commercial eyetracking recording and analysis software. The software records eye movements, eye fixations, as well as webpage access, mouse events and keystrokes.

- *Morae* is a commercial software package for usability testing and user experience developed and distributed by TechSmith[3]. It records participants' webcam and computer screen as video, captures audio, and logs screen text, mouse clicks and keystrokes occurring within Internet Explorer.

This extensible list of logging tools are loosely coupled to the *Interaction Logger* and the *Remote Logging Application Framework* components through task configurations for individual tasks. The task configuration describes which logging tools are used during a task and the software framework activates them as soon as participants enter a task and deactivates them as soon as they complete a task.

The researcher can create a selection of relevant tools for each task of a particular IIR experiment from the available logging tools supported by the system framework. First, one should select all user behavior the researcher is interested in. Second, the observable data types that provide evidence for the existence and the structure of these user behaviors is identified. Finally, these data types are linked with relevant logging tools. In the next section we summarize experiences from three distinct experiments that were designed and performed with our experiment system framework. We do not describe these experiments in this paper. Instead, we focus on key points and issues that should be addressed when collecting multidimensional logging data from hybrid logging tools.

4. EXPERIENCES FROM MULTIDIMENSIONAL DATA LOGGING

Data logging with an array of hybrid tools, as described in the previous section, has a number of benefits and challenges. This section summarizes our initial experiences from conducting three IIR user experiments with the system framework and some initial processing and integration of its data logs.

[2]http://www.tobii.com
[3]http://www.techsmith.com

- *Accuracy and Reliability:* Using data streams from multiple logging tools limits the risk of measurement errors to enter data analysis. This is especially relevant to IIR due to its need to conduct experiments in naturalistic settings where people perform tasks in conditions that are not fully controlled and therefore less predictable. Such settings allow participants to solve tasks with great degrees of freedom. As a result of this, user actions in such settings tend to be highly variable. Measurement errors or missing data, for example based on varying system performance and network latencies, have a larger impact because the entire interaction is studied. Multiple data streams from different sources improve the overall accuracy of recorded sessions and increase the reliability of detecting features in individual logs. Furthermore, the use of multiple data logs limits of chances that artifacts created by individual logging tools and their assumptions will affect downstream analysis.

- *Disambiguation:* The use of multiple data logs allows to contextualize each log with the logs produced by other tools and disambiguate uncertainties in the interpretation of logging event sequences. We found that the most common cases are timestamp disambiguation and the synchronization of event accuracies.

  - *Timestamp disambiguation:* The timestamp granularity of recorded events usually varies between logging tools. For example, Tobii Studio records eye tracking data with a constant frequency determined by the eye tracking hardware (e.g. 60 logs per second (17 ms) for the T60 model) whereas UsaProxy records events only every full second and RUIConsole records events dynamically only when they occur. The combination of logging data from different tools helps to better determine the real timing of events by providing different viewpoints for the same sequence of actions a user has performed. Low granularity timestamps might collapse a number of user events to a single point of time and, based on that, change the natural order in which these events are recorded. Alternative secondary logging data can help to detect such event sequences and help disambiguating and correcting them.
  - *Detail of event structure:* Every logging tool imposes a number of assumptions on the data produced by a user – which events to log, which events to differentiate and how to label them. Two logging tools recording the same events can therefore produce different event structures with varying detail. For example, RUIConsole differentiates a mouse click into a press and a release event whereas Tobii Studio considers a mouse click as a single event. Different logging tools recording the same user actions produce events with a structure of different detail that can be used to contextualise conflicting recordings of user actions.

- *Scalability:* Concurrent use of logging tools may create performance issues on the client machine especially with tools that produce large amounts of data. Especially the combined use of Morae and Tobii Studio can be demanding when using high quality web camera and screen capture recording. Limited hardware resources may have a direct effect on the recording accuracy of other logging tools. More importantly, however, a overloaded client may have an effect on participants and their ability to accomplish tasks realistically. This can be avoided by choosing a sufficiently equipped client machine and a fast network. As mentioned in section 4, the software framework supports the distribution of logging tools over several machines, while these tools are activated centrally by the server architecture, which can help to better balance the load.

- *Stability:* Concurrent use of multiple logging applications can destabilize the client computer. Individual applications can affect each other especially when logging from the same resources (e.g. from the same instance of Internet Explorer). Currently, our system framework does not monitor running logging tools and there is no mechanism to recover tools that hang or break during a task. This is a feature we will incorporate into a future version of the system framework.

5. FUTURE WORK

Future work on the experiment system framework will focus on further improvement of logging tool integration and monitoring. We are currently developing a graphical user interface for researchers to more easily design IIR experiments with the system and monitor progress of running experiments and the accuracy of its data logs. An extension to the experiment system framework presented in this paper is a data analysis system that allows us to fully integrate, analyse and develop models from the recorded data. In particular, we are interested in creating higher level constructs from integrated low-level logging data that can be used to personalise interactive search for users. The experiment system framework will be released as open source to the wider research community.

6. REFERENCES

[1] R. Atterer, M. Wnuk, and A. Schmidt:. Knowing the User's Every Move - User Activity Tracking for Website Usability Evaluation and Implicit Interaction. In *15th International World Wide Web Conference (WWW2006)*, Edinburgh, Scotland, 2006.

[2] N. Belkin. Intelligent Information Retrieval: Whose Intelligence? In *Fifth International Symposium for Information Science (ISI)*, pages 25–31, Konstanz, Germany, 1996. Universtaetsverlag Konstanz.

[3] N. Belkin, C. Cool, A. Stein, and U. Thiel. Cases, Scripts, and Information-Seeking Strategies: On the Design of Interactive Information Retrieval Systems. *Expert Systems with Applications*, 9(3):379–395, 1995.

[4] A. Edmonds, K. Hawkey, M. Kellar, and D. Turnbull. Workshop on logging traces of web activity: The mechanics of data collection. In *15th International World Wide Web Conference (WWW 2006)*, Edinburgh Scotland, 2006.

[5] U. Kukreja, W. E. Stevenson, and F. E. Ritter. RUI – Recording User Input from interfaces under Windows and Mac OS X. *Behavior Research Methods*, 38(4):656–659, 2006.

Incorporating user behavior information in IR evaluation

Emine Yilmaz Milad Shokouhi Nick Craswell Stephen Robertson
Microsoft Research Cambridge
Cambridge, UK
{eminey, milads, nickcr, ser}@microsoft.com

ABSTRACT

Many evaluation measures in Information Retrieval (IR) can be viewed as simple user models. Meanwhile, search logs provide us with information about how real users search. This paper describes our attempts to reconcile click log information with user-centric IR measures, bringing the measures into agreement with the logs. Studying the discount curve of NDCG and RBP leads us to extend them, incorporating the probability of click in their discount curves. We measure accuracy of user models by calculating 'session likelihood'. This leads us to propose a new IR evaluation measure, Expected Browsing Utility (EBU), based on a more sophisticated user model. EBU has better session likelihood than existing measures, therefore we argue it is a better user-centric IR measure.

1. INTRODUCTION

This paper is concerned with user-centric IR evaluation, where an evaluation measure should model the reaction of a real user to a list of results, evaluating the utility of the list of documents to the user. Web search experiments usually employ an IR measure that focuses only on top-ranked results, under the assumption that Web users deal 'shallowly' with the ranked list. This is probably correct, but we might ask: How can we be sure that Web search users are shallow, and how should we choose the degree of shallowness. In this paper, our solution is to make IR evaluation consistent with real user click behavior. We still evaluate based on relevance judgments on a list of search results, but the importance of each search result is brought in line with the probability of clicking that result.

In our experiments we use click logs of a search engine (`bing.com`) taken from January 2009, combined with relevance judgments for 2057 queries. For each judged query we extracted the top-10 results for up to 1000 real query instances, and the pattern of clicks in the form of 10 Booleans (so each result is either clicked or not clicked). More than 91% of all top-10 query-URL pairs were judged on the 5-level scale {Perfect, Excellent, Good, Fair, Bad}. Unjudged documents are assumed to be Bad. We divide the queries into two sets of equal size: training and test.

A key difference between user-centric IR evaluation measures, such as Normalized Discounted Cumulative Gain (NDCG) [2] and Rank Biased Precision (RBP) [3], is the choice of discount function. Many experiments with NDCG apply a discount at rank r of $1/\log(r+1)$. Another metric, RBP, has a persistence parameter p so that the probability of seeing position r is p^{r-1}. Note, some evaluation measures such as Average Precision are not easily interpretable as a user model. Such measures are beyond the scope of this paper, since we focus on user-centric evaluation.

The next section considers the discount curves of NDCG and RBP, in contrast to real click behavior. Noting a discrepancy, we extend the two metrics based on information about the probability of click on each relevance label. Having done so, the discount curves are more in line with real user behavior. However, the curves do not incorporate information about the user's probability of returning to the results list, having clicked on a result. Therefore the next section introduces our new evaluation measure Expected Browsing Utility (EBU). Finally we introduce Session Likelihood, a test for whether an evaluation measure is in agreement with click logs. Under that test, EBU is most in line with real user behavior, therefore we argue it is a superior user-centric evaluation measure.

2. DISCOUNT FUNCTIONS AND CLICKS

One of the key factors for differentiating between the evaluation metrics is their *discount functions*. Most user-centric IR evaluation metrics in the literature can be written in the form of $\sum_{r=1}^{N} p(\text{user observes document at rank r}) \cdot gain(r)$ as the discount function is assumed to be modeling the probability that the user observes a document at a given rank. Therefore, the quality of a metric is directly dependent on how accurately the discount function estimates this probability. In the case of Web search, this probability value should ideally correspond to the probability that the user *clicks* on a document at rank r. Hence, one can compare the evaluation metrics based on how their discount function (their *assumed* probability of click) compare with the actual probability that the user clicks on a document. Discount functions that are more consistent with click patterns are more flexible in explaining – and evaluating – the users Web search behavior.

Next, we compare the user models associated with the underlying discount functions of RBP and NDCG. The top two plots in Figure 1 show the average probability of click (averaged over all sessions in the test data) per rank. We then compare this actual probability of click with the click probability *assumed* by different evaluation metrics. As mentioned above, this probability corresponds to the *discount* function used in the definition of the metrics. The upper

SIGIR'09, July 19-23, 2009, Boston, USA.

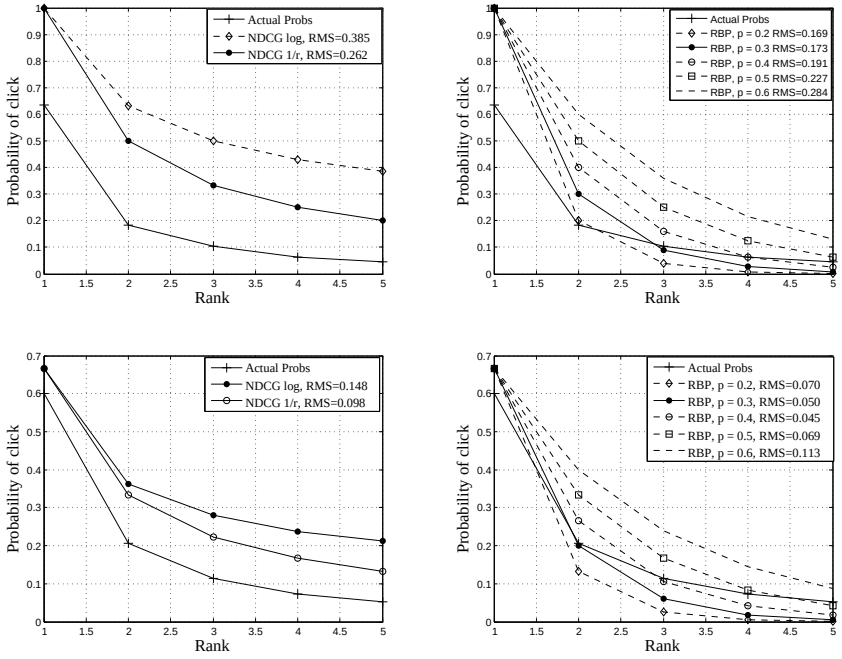


Figure 1: P(click) vs. rank for different metrics.

left and right plots compare the discount function of NDCG (with the commonly used $1/\log_(r+1)$ and $1/r$ discounts) and RBP (with $p \in \{0.2, 0.3, 0.4, 0.5, 0.6\}$) with the actual click probability, respectively. For comparison purposes, the plots report the Root Mean Squared (RMS) error between the probability of click assumed by a metric and the actual probability of click. It can be seen that the probability of click assumed by these two metrics is quite different than the actual click probability.

As the discount functions in NDCG and RBP are not derived from search logs, it is not surprising to see that they are not successful in predicting clicks. In the following section, we show how extending such metrics by incorporating the quality of snippets can significantly improve the discount functions for predicting the probabilities of clicks.

3. MODELING THE IMPACT OF SNIPPETS

One reason for the discrepancy between the described discount functions and the click patterns is that these metrics do not account for the fact that the users only click on *some* documents depending on the relevance of the summary (snippets). Both RBP and NDCG assume that the user *always* clicks on the document at the first rank, whereas the actual probability of click calculated from our training search logs shows that the probability that the user clicks on the first ranked document is only slightly higher than 0.6.

To address this issue, we enhance the NDCG and RBP user models by incorporating the snippet quality factor and considering its impact on the probability of clicks. We hypothesize that the probability that the user clicks on a document (i.e., the quality of the summary) is a direct function of the relevance of the associated document. Table 1 supports our claim by showing $p(C|summary) \sim p(C|relevance)$ obtained using the training dataset.[1]It can be seen that the probability that the user clicks on a document tends to increase as the level of relevance of the document increases. Note that this behavior is slightly different for *Bad* and *Fair* documents, in which case there is a slight difference in the click probability. This is caused by the fact that (1) the documents judged as *Fair* tend to be slightly relevant to the user information need; hence, they are *effectively Bad* to the user, and (2) the unjudged documents are treated as *Bad* in our computations.

Table 1: Probability of click given the relevance

Relevance	$P(click\|relevance)$
Bad	0.5101
Fair	0.5042
Good	0.5343
Excellent	0.6530
Perfect	0.8371

Motivated by these observations, we extend NDCG and RBP to incorporate the *summary* quality into their discount functions as follows: If the discount function of the metric dictates that the user visits a document at rank r with probability $p(d_r)$, then the probability that the user *clicks* on the document at rank r can be computed as $p(d_r) \cdot p(C|summary_r)$ (where the click probabilities are shown in Table 1). The bottom two plots in Figure 1 show how the extended versions of metrics then compare with the actual click probability. It can be seen that the extended versions

[1]For simplicity, we assume that the quality of summaries and the relevance of documents are strongly correlated. That is, relevant summaries for relevant documents and vice versa.

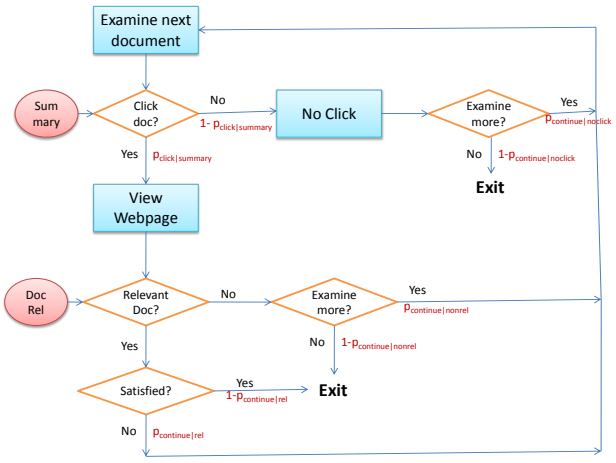


Figure 2: The user browsing model associated with the new evaluation metric.

Table 2: Probability of continue given the relevance

Relevance	$P(cont\|relevance_r)$
Bad	0.5171
Fair	0.5727
Good	0.6018
Excellent	0.4082
Perfect	0.1903

of these metrics can approximate the actual probability of click substantially better than the standard versions.

We would like to note that Turpin et al. [4] recently also suggested that document summary information should be incorporated in evaluation retrieval evaluation, independent of our work. They showed that using the summary information in evaluation may alter the conclusions regarding the relative quality of search engines. However, their work mainly focus on average precision as the evaluation metric.

4. EXPECTED BROWSING UTILITY (EBU)

All the metrics described so far assume that the probability that the user will continue search at each rank is independent of (1) whether the user has clicked on a document or not, and (2) the relevance of the document seen by user. Intuitively, we expect the search behavior of users to change based on the relevance of the last visited document. That is, visiting a highly relevant document that perfectly satisfies the user's information need (e.g. a navigational answer) shall be strongly correlated with the probability of terminating the search session.

We confirmed our hypothesis by computing the probabilities of *continuing* the search session conditioned on the relevance of the last clicked document. The results generated from our training set are shown in Table 2. It can be seen that if the document is very relevant to the information need (e.g., *Perfect*), then the user is likely to stop browsing the results as he has found the information he was looking for. On the other hand, if the user clicks on a document that is not relevant to his information need (e.g., *Bad*), then he is again likely to stop browsing as he is frustrated with the result he has clicked on and thinks documents retrieved lower than that will probably be even less relevant.

Motivated by the probabilities of click and continue shown in Tables 1 and 2, we propose a novel user model in which: (1) When a user visits a document, the user may or may not click the document depending on the quality of the summary, and (2) The relevance of a document visited by a user directly affects whether the user continues the search or not.

Figure 2 shows the user model associated with our metric. The associated user model can be described as follows: The user starts examining the ranked list of documents from top to bottom. At each step, the user first just observes the *summary* (e.g., the snippet and the url) of the document. Based on the quality of the summary, with some probability $p(C|summary)$ the user clicks on the document. If the user does not click on the document, then with probability $p(cont|noclick)$ he/she continues examining the next document or terminates the search session with probability $1 - p(cont|noclick)$.

If the user clicks on the document, then he or she can assess the *relevance* of the document. If the document did not contain any relevant information, then the user continues examining with the probability $p(cont|nonrel)$ or stops with $1 - p(cont|nonrel)$ probability. If the clicked document was relevant, then the user continues examining with probability $p(cont|rel)$ (which depends on the relevance of the clicked document).

A similar user model has been suggested by Dupret et al. [1]. However, their work is mainly focused on predicting the future clicks, while our goal is to integrate the probabilities of clicks with evaluating the search results.

We use past click data together with relevance information to model the user search behavior. At each result position r, our model computes the expected probability of examining the document $p(E_r)$ as follows: We first assume that the user always examines the very first document, hence $p(E_1) = 1$. Now, suppose the user has examined the document at rank $r-1$ and we would like to compute $p(E_r)$. Given that the user has already examined the document at $r-1$, according to our model, with probability $p(C|summary_{r-1})$ the user clicks on the document at rank $r-1$, observes the relevance of the document at rank $r-1$ and continues browsing the ranked list with probability $p(cont|rel_{r-1})$. Alternatively, with probability $1 - p(C|summary_{r-1})$ the user does not click on the document at rank $r-1$ and continues browsing with probability $p(cont|noclick)$. Overall, the probability that the user will examine the document at rank r can be written as:

$$\begin{aligned} p(E_r) &= p(E_{r-1}) \cdot [p(C|summary_{r-1}) \cdot p(cont|rel_{r-1}) \\ &+ (1 - p(C|summary_{r-1})) \cdot p(cont|noclick)] \end{aligned}$$

Given that the user has examined the document at rank r, the probability that the user clicks on this document is $p(C|summary_r)$. That is, the user clicks on a document at rank r with probability $p(C_r) = p(E_r) \cdot p(C|summary_r)$.[1]

Therefore, in total, the **Expected Browsing Utility (EBU)** that the user receives from the output of the search engine is then $EBU = \sum_{r=1}^{N} p(C_r) \cdot rel_r$ (divided by the EBU value of an optimal list so that the metric is between 0 and 1), where rel_r is the relevance of document at rank

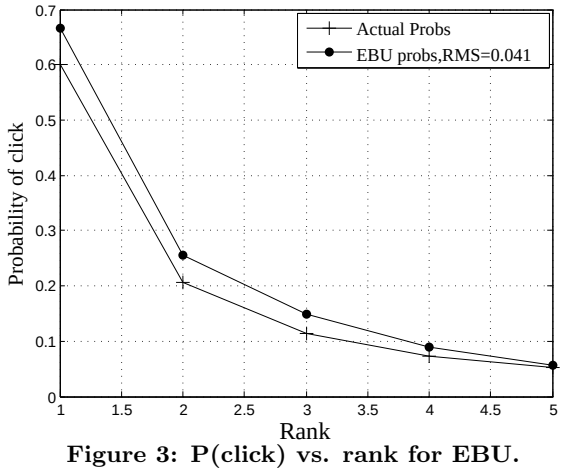


Figure 3: P(click) vs. rank for EBU.

r. In EBU, the importance of a document d depends on (1) its relevance and (2) the probability that d is clicked and viewed by the user.

Figure 3 shows the same curves using EBU as the metric (computed using probabilities from Table 1 and Table 2). Comparing the EBU curves with those in Figure 1, it can be seen that EBU is better than both versions of NDCG and RBP.

5. EVALUATING EVALUATION METRICS

In the above experiments we focused on the *average* click probability, i.e., the average probability that a user will click on a document at some rank r. Ideally, one would like to be able to infer the individual clicks per session. This way, the evaluation of user satisfaction per user session would be much accurate. Hence, in the second set of experiments, we compare the probability of click dictated by the discount function of a metric with the actual click observations per session.

For that, we use the click probability dictated by an evaluation metric as a generative model and then compute the probability that this distribution would generate the sessions that were observed in the test data (i.e., the *session likelihood*). Instead of computing the session likelihood, one can also compute the session log likelihood. Let $p(C_r|M)$ be the probability of click at rank r dictated by the discount function of the metric M and let the likelihood of a particular session s given this metric be

$$P(s|M) = \prod_{\forall r, doc_r \in C_s} P(C_r|M) \cdot \prod_{\forall r, doc_r \in NC_s} (1 - P(C_r|M))$$

where C_s and NC_s correspond to the documents clicked and not clicked in session s, respectively and doc_r refers to the document at rank r in session s. The session log likelihood can then be written as:

$$\begin{aligned} log(P(sessions|M)) &= \log[\prod_{\forall s \in sessions} P(s|m)] \\ &= \sum_{\forall s \in sessions} \log(P(s|m)) \end{aligned}$$

The first column in Table 3 shows the session log likelihood for each metric. For comparison purposes, the second column in the table shows the average probability of observing the sessions in the test data. It can be seen that EBU can predict the behavior of an individual user (i.e., per session) much better than all the other metrics.

	Session Log Likelihood	P(click per session)
RBP, p=0.2	-2.3859	0.0920
RBP, p=0.3	-2.1510	0.1164
RBP, p=0.4	-2.0570	0.1278
RBP, p=0.5	-2.0732	0.1258
RBP, p=0.6	-2.2007	0.1107
NDCG, log	-2.3064	0.0996
NDCG, 1/r	-2.0435	0.1296
EBU	-1.9371	0.1441

Table 3: Likelihood of individual sessions given each evaluation metric.

6. CONCLUSIONS

Most evaluation metrics in information retrieval aim at evaluating the satisfaction of the user given a ranked list of documents. Hence, these metrics are based on some underlying user models which are assumed to be modeling the way users search. However, most of these user models are based on unrealistic assumptions.

In this paper, we showed how click logs can be used to devise enhanced evaluation measures. We first extended two commonly used evaluation metrics, NDCG and RBP, to incorporate probability of click in their discount curves. We then introduced EBU, new evaluation metric that comes from a more sophisticated user model than the other metrics. Finally, using a novel evaluation methodology of evaluating evaluation measures (referred to as *session likelihood*), we compared these different metrics and showed that EBU is a better metric in terms of modeling user behavior.

7. REFERENCES

[1] G. E. Dupret and B. Piwowarski. A user browsing model to predict search engine click data from past observations. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 331–338, Singapore, Singapore, 2008. ACM.

[2] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002.

[3] A. Moffat, W. Webber, and J. Zobel. Strategic system comparisons via targeted relevance judgments. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 375–382, Amsterdam, The Netherlands, 2007. ACM.

[4] A. Turpin, F. Scholer, K. Jarvelin, M. Wu, and J. S. Culpepper. Including summaries in system evaluation. In *SIGIR '09: Proceedings of the 32nd annual international ACM SIGIR conference on Research and development in informaion retrieval*, Boston, MA, USA, 2009. ACM. To Appear.

Faceted Search for Library Catalogs: Developing Grounded Tasks and Analyzing Eye-Tracking Data

Robert Capra[1], Bill Kules[2], Matt Banta[2], Tito Sierra[3]

rcapra3@unc.edu, kules@cua.edu, matt.banta@gmail.com, tito_sierra@ncsu.edu

[1]School of Information and Library Science
University of North Carolina
Chapel Hill, NC

[2]School of Library and Information Science
The Catholic University of America
Washington, DC

[3]NCSU Libraries
North Carolina State University
Raleigh, NC

ABSTRACT

In this paper, we describe two aspects of a study we conducted of faceted search in an online public access library catalog (OPAC). First, we describe how we used log data from a university OPAC to develop a set of grounded tasks. Then, we describe our use of eye-tracking in a controlled laboratory setting to examine user behaviors performing the grounded tasks. We discuss the challenges we encountered both in using the log data to develop tasks and in collecting and analyzing the eye-tracking data.

Categories and Subject Descriptors

H.5.2 User Interfaces: Evaluation/methodology; H.3.3 Information Search and Retrieval

General Terms

Measurement, Experimentation, Human Factors

Keywords

Eye-tracking, log file analysis, library catalog, OPAC, faceted search

1. INTRODUCTION

Many libraries have recently redesigned their online public access catalogs (OPACs) to include faceted metadata as part of the search interface. In these systems, metadata such as the Library of Congress subject headings, time period, and region are displayed as facets that can be used to explore and refine search results (see Figure 1). There are many open research questions about how people use facets in a search process and the library science community is especially interested in how these redesigned OPACs are being used. We designed a study to examine how long and in what sequences searchers looked at the major elements of a faceted OPAC interface [2].

This paper describes two types of challenges encountered along the way: developing exploratory search tasks and analyzing eye tracking data.

2. LOG ANALYSIS OF SEARCHES

Our study needed search tasks that balanced two competing needs: first, the tasks needed to induce an exploratory mode of search instead of the directed mode used in many studies. Second, the tasks needed to be constructed in a way that allowed us to make comparisons between subjects. In addition, the tasks needed to be appropriate for the catalog available on the test system, which was based on the North Carolina State University (NCSU) Libraries OPAC, reflecting real usage of that catalog. The online library catalog for NCSU serves on average 7,824 search transactions and 1,087 user sessions per day [1].

To develop the tasks, we extracted three days of anonymized log data from the servers. This extracted data included both keyword search terms and any facets used in the searches. Our goal was to use this log data to identify actual searches executed on the NCSU OPAC that made use of facets. We were especially interested in identifying exploratory searches (as opposed to directed or known-item searches) in the log data.

We manually looked through the extracted log data to identify situations in which the user appeared to be doing an exploratory search that also included the use of facets. We looked for log entries where it was clear that the user issued several searches with the same or related keywords and in which they interacted with the results. Our selection criteria required that the log file show that the searcher: 1) had looked through more than one page of results, 2) had selected more than one facet that was not identical to the search term, and 3) the selected facets were from the subject, time period, and region facets. The deployed NCSU OPAC has additional facets, but we decided to focus on only these three for our study.

To further define the tasks, we then conducted our own searches using the topics that were extracted from the log files. If a single keyword search could easily address the topic, it was either rejected as too easy or modified to either broaden or narrow its scope. Iterating this process, we developed a set of four exploratory search tasks to use in the study. More details of our task development and refinement process are given in [2].

There are obvious difficulties in isolating exploratory searches from log data. First, the log data did not link queries across sessions, so there was no way of knowing with 100% certainty that two queries were done by the same user. However, we often observed sequences of closely related search terms in close time proximity that indicated an exploratory style search. Second, it is often impossible to know the exact motivations behind the actions observed in the log data. For example, what was the underlying task that lead a searcher to issue the query? Why did they chose to click on that facet? However, for our purposes, the log data provided a rich set of indicators to use in developing a set of exploratory search tasks grounded in real-world searches.

SIGIR'09, July 19-23, 2009, Boston, USA.

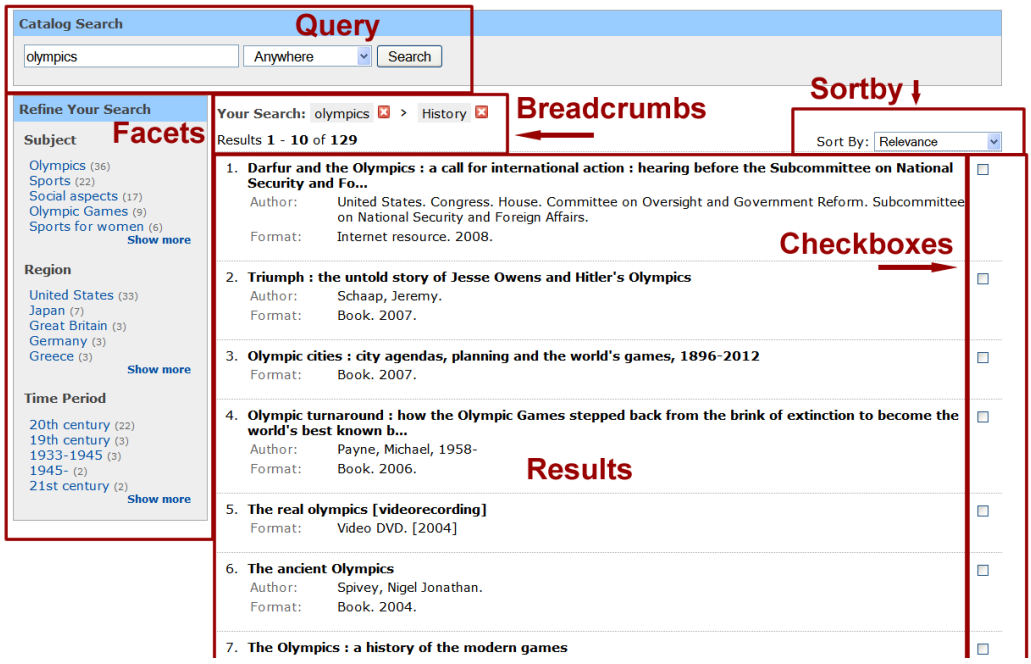


Figure 1. Faceted OPAC Interface showing six areas of interest (AOIs)

3. COLLECTING EYE-TRACKING DATA

3.1 Interface Design

We used a Tobii 2150 remote eye tracker (http://www.tobii.com) to collect the eye-tracking data. This system includes a 21" LCD monitor with embedded infrared cameras that sample at 50Hz. The monitor resolution was set to 1024x768.

For this study, we focused on how facets were used in the search process. The deployed NCSU OPAC includes many interface elements and features that are tangential to our current research interest. To keep the study focused, we developed a customized OPAC interface, shown in Figure 1. There were six major areas of interest (AOIs) in the interface: 1) a keyword query search box, 2) an area to display the facets, 3) a breadcrumb trail showing the current search terms and selected facets, 4) an area to display the results list, 5) a drop-down menu to select how to sort the results, and 6) a checkbox for each result so that the user could indicate which results they wanted to record as their "answers" for each task. This customized interface still accessed the full NCSU catalog of over 1.8 million records.

To facilitate collecting the eye-tracking data, we made several adjustments to this customized interface. First, we made sure that the interface used fixed-width elements when possible so that we could easily define a template for the areas of interest on each page. Second, we included 5 pixels of "padding" between interface elements to help increase the precision of gaze data collection for specific AOIs.

3.2 Data Collection

Data collection using the eye-tracker was a tricky process. First, we seated each participant at the computer with the eye-tracker and went through a calibration process. After the first and second tasks, while the participant was completing a post-task questionnaire, the experimenter would quickly skim a video that showed the eye-traces that were captured from the previous task to make sure that the eye-tracking was good. In cases where it had problems, we would either recalibrate the equipment and/or remind the participant to sit as they had been sitting when doing the calibration. For two participants, the equipment could not maintain tracking for more than a few seconds and we had to discard the tracking data.

We observed that changes of posture were often the cause of eye-tracking failure. A typical example was that participants would sit in a neutral posture while doing the calibration, but then either slump or "lean in" while engaged in the tasks. We often had to gently remind participants during the tasks to resume their original posture. While we initially were reluctant to interrupt them to correct their posture, we believe that the negative impact of this interruption was very small compared to the gains in better eye-tracking. We often used wording to encourage the participant to help us, such as, "The equipment is being finicky today, could you just sit up a bit so it can track you better?" Other types of eye-tracking such as head-mounted units might not have these issues with posture causing a loss of tracking.

The challenge of maintaining tracking has encouraged us to consider using a secondary monitor that will display the tracking status in our subsequent studies. This will allow us to monitor the tracking in real-time during the tasks and to encourage the participants to adjust their posture if needed.

One other challenge encountered was caused by the automatic update feature of Microsoft Windows. During the course of data collection (which spanned a week), the system performed an automatic update which upgraded the Internet Explorer browser to version 7. This was not compatible with the Tobii eye tracker and forced us to reschedule several sessions while we downgraded back to IE6.

4. ANALYZING EYE-TRACK DATA

Tobii Clearview analysis software (v 2.7.1) was used to segment each web page viewed into the areas of interest (AOIs). This was a labor intensive step. Each web page viewed had to be segmented by hand by defining a box around each AOI using a GUI tool. Templates can be used to define the locations of fixed size and fixed position AOIs. However, for each page, the AOIs from this template had to be adjusted because some of the interface elements were of variable size (both horizontal and vertical). For example, the vertical size of the facet AOI depended on the number and length of the facets. Cutrell et al. [3] overcame a similar problem by embedding custom JavaScript code in the web pages they were studying that automatically extracted the locations and dimensions of bounding boxes based on the Document Object Model (DOM) of the page. These dimensions could then be used to automatically generate the AOIs definitions.

We analyzed the raw eye-gaze data to extract fixations that had a minimum of 100ms duration within a radius of 30 pixels. Different domains use different fixation criteria. For example, for reading text, fixations may be more tightly defined than for image-oriented tasks such as visual search. For reading tasks, the manufacturer (Tobii) recommended a 20 pixel radius for 40ms. For image tasks, they recommend a 50 pixel radius for 200ms. Because our tasks involved both aspects of reading and visual search, we chose their recommendations for mixed content (30 pixel radius for 100ms duration).

After defining the AOIs and extracting the fixations, the Tobii software output a time ordered sequence of gaze data. We wrote scripts in PHP to convert and analyze this data. The scripts had to accumulate fixations across AOIs, tasks and individual page views.

In analyzing eye-tracking data, two measures have been widely used for related studies: fixation counts and fixation times. Fixation count is thought to be an indicator of the importance of the item (or AOI) being fixated upon [4]. Fixation time is considered to be an indicator of the complexity of the element. We initially focused on analysis of the cumulative fixation time for each AOI, but became interested in the transitions between AOIs to examine the pattern of eye movement on the page. Specifically, we extracted "gaze transition pairs" between AOIs for all participants, task scenarios, and page views. We used this data to generate directed graphs to summarize the most commonly occurring gaze paths between AOIs. An example graph is shown in Figure 2. This technique allowed us to see that many transitions occurred between the results and facets and between the results and breadcrumb area. We believe that directed graph summarization shows great promise as an eye-tracking data analysis tool.

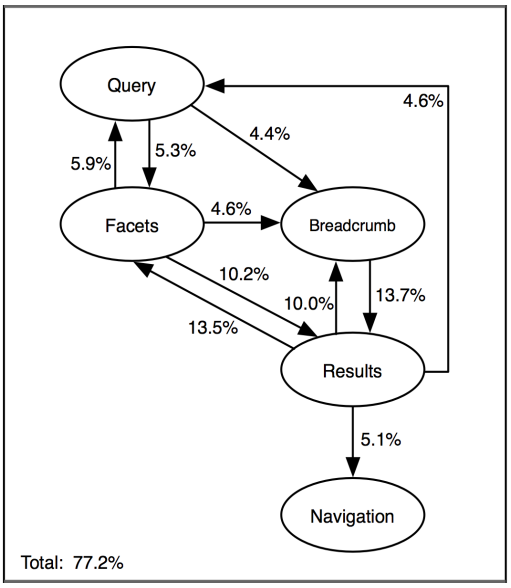


Figure 2. Example Gaze Transition Directed Graph

5. ACKNOWLEDGMENTS

This work was supported in part by grants from the NSF/Library of Congress (ISS 0455970 and ISS 0812363) and a grant from the Catholic University Grant-in-Aid Committee. We thank Doug Oard for the use of the eye tracker, and Joseph Ryan and Jason Casden for their help in configuring the interface for this study.

6. REFERENCES

[1] Lown, C. (2008). A Transaction Log Analysis of NCSU's Facted Navigation OPAC. Master's paper, School of Information and Library Science, University of North Carolina at Chapel Hill, 2008.

[2] Kules, B., Capra, R., Banta, M., Sierra, T. (2009). What Do Exploratory Searchers Look at in a Faceted Search Interface? In *Proceedings of the 9th ACM/IEEE-CS Joint Conference on Digital libraries (Austin, TX, June 15 - 19, 2009)*. JCDL 2009. ACM Press, New York, NY.

[3] Cutrell, E. and Guan, Z. 2007. What are you looking for?: an eye-tracking study of information usage in web search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (San Jose, California, USA, April 28 - May 03, 2007). CHI '07. ACM, New York, NY, 407-416.

[4] Jacob, R. & and Karn, K. (2003) Eye Tracking in Human-Computer Interaction and Usability Research: Ready to Deliver the Promises (Section Commentary), in *The Mind's Eye: Cognitive and Applied Aspects of Eye Movement Research*, ed. by J. Hyona, R. Radach, and H. Deubel, pp. 573- 605, Amsterdam, Elsevier Science.

How Task Types and User Experiences Affect Information-Seeking Behavior on the Web: Using Eye-tracking and Client-side Search Logs

Hitomi SAITO
Aichi University of Education
1 Hirosawa, Igaya-cho,
Kariya-shi, Aichi, Japan
hsaito@auecc.aichi-edu.ac.jp

Hitoshi TERAI
Tokyo Denki University
2-1200 Muzai Gakuendai,
Inzai-shi, Chiba, Japan
terai@sie.dendai.ac.jp

Yuka EGUSA
National Institute for Educational Policy Research
3-2-2 Kasumigaseki,
Chiyoda-ku, Tokyo, Japan
yuka@nier.go.jp

Masao TAKAKU
National Institute for Materials Science
1-2-1 Sengen, Tsukuba–shi,
Ibaraki, Japan
TAKAKU.Masao@nims.go.jp

Makiko MIWA
The Open University of Japan
2-11 Wakaba, Mihama, Chiba, Japan
miwamaki@code.u-air.ac.jp

Noriko KANDO
National Institute of Informatics
2-1-2 Hitotsubashi,
Chiyoda-ku, Tokyo, Japan
kando@nii.ac.jp

ABSTRACT

We investigated what influence task type and user experience had on information-seeking behaviors on the Web by using screen-capture logs and eye-movement data. Five graduate students in library and information science and eleven undergraduate students with other majors performed two different Web searches, a report-writing and a trip-planning task, and their think-aloud protocols, behaviors, and eye movements were recorded. Analyses of the screen-capture logs and eye-movement data revealed that the task type and user experience affected the participants' information-seeking behaviors.

1. INTRODUCTION

Originally developed as a means of searching the Web for information, search engines have become fairly routine and increasingly important in our everyday lives [6]. Considerable research has been done using a variety of methodologies, e.g., analysis of search-engine logs, user experiments, questionnaires, and interviews, to determine how ordinary people use search engines. Because searching for information on the Web is a process of browsing through individual Web pages that are offered by a search engine in response to a query, the ability to support exploratory searches is crucial [4]. This motivated us to pursue quantitative user trials and experiments with the goal of clarifying the exploratory search process by collecting various data from a pre-test questionnaire, client-side search logs, think-aloud protocols, eye-tracking, and post-experiment interviews [7].

One of the main objectives of this study is to deepen our understanding of the relationship between search behavior and the characteristics of different tasks. A number of studies have examined differences in search behavior in dealing with different tasks [3, 8]. In this study, we compare a report-writing task with a trip planning task. These tasks correspond, respectively, to informational and transactional in Broder's taxonomy [1].

We also studied how different levels of knowledge and experience affected the search behaviors of participants conducting exploratory searches. We compared the search behaviors of undergraduate students of various majors with those of graduate students of library and information science. There have been many studies examining the effects of experience on search behaviors [5]. Yet, very few of these studies have analyzed the kind of information that users are searching for. We used eye-tracking data to analyze what students were looking at on the screen, and we then determined whether these viewing tendencies were correlated with differences in experience. The following sections detail our experimental methodology and analytical findings.

2. METHODOLOGY

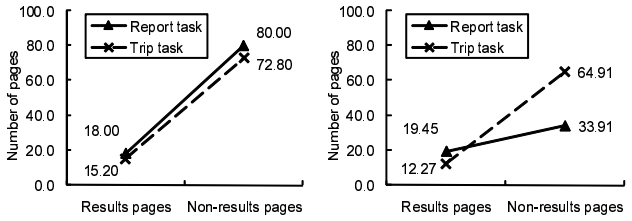
2.1 Participants and Tasks

The participants were 11 undergraduate (ages: between 19 and 21; male: 5, female: 6) and 5 graduate students (ages: between 23 and 28; male: 4, female: 1). The undergraduate students' academic majors included economics, literature, electronics engineering, Spanish, psychology, chemistry, and civil engineering, and the graduate students' were in library and information science.

Two groups differed in terms of web browsers and search engines that they used. Most of the undergraduate students used Internet Explorer 6 (IE6: 10, Firefox: 1). In contrast, almost all of the graduate students used tab browsers (Sleipnir: 2, Firefox: 1, Opera: 1, Others: 1). Almost half of the undergraduate students used Yahoo!Japan as their search engines; the graduate students all used Google.

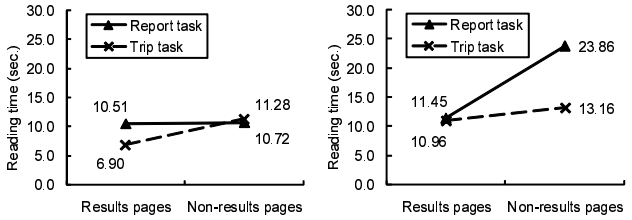
The participants were requested to conduct two different Web searches: a report-writing (report task) and a trip-planning task (trip task). They selected a particular topic for each task based on their own interests because we wanted their search to be exploratory in nature.

SIGIR'09, July 19-23, 2009, Boston, USA.



(a) Graduate students (b) Undergraduate students

Figure 1: Average number of results pages and non-results pages viewed



(a) Graduate students (b) Undergraduate students

Figure 2: Average viewing time for results pages and non-results pages

2.2 Procedure and methods

The participants answered questions in a pre-test questionnaire about their information-seeking experience with Web-search engines. They were instructed to use their favorite search engine in the experiment. They were given a five-minute period to conduct a Web search and practice the "think-aloud" method, in which they orally described their thought processes. Two experimental search tasks (report and trip tasks) were then conducted for fifteen minutes. The order of the searches was counterbalanced between participants. Their eye-movements during the experiments were recorded with an eye-tracking system (EMR-AT VOXER, NAC Image Technology Inc.). They were required to think aloud, and the log data were recorded.

After each search, the participants completed a questionnaire about the degree of difficulty and satisfaction with their searches. We subsequently interviewed them about their information-seeking process while watching screen-capture video of their PC use together with eye movements to facilitate episodic memory retrieval.

3. RESULTS AND DISCUSSION

We next report the results of analysis based on the browser logs, screen-capture video, and the eye-movement data.

3.1 Behavioral Data Analysis

Analysis of Number of Pages and How Long They Were Viewed

We analyzed the number of pages participants viewed and how long they were viewed for two types of tasks and two groups. The pages were classified into two types: results and non-results pages. The results pages were of results or hits that were presented by the search engine in response to queries, and the non-results pages were Web pages other than these.

Table 1: Number of search actions per task

Categories of Action	Report-writing task				Trip-planning task			
	Graduates		Undergraduates		Graduates		Undergraduates	
	Mean	*SD*	Mean	*SD*	Mean	*SD*	Mean	*SD*
Search	9.20	3.35	8.00	4.63	7.80	5.89	6.36	5.18
Link	25.80	14.13	19.18	6.66	29.00	14.47	35.73	9.12
Next	0.80	0.84	0.36	0.81	0.20	0.45	0.91	1.14
Back	10.40	9.07	17.18	7.97	10.80	8.04	22.00	14.61
Jump	2.20	1.92	2.36	1.69	3.20	2.68	2.45	1.75
Browse	0.80	1.30	1.82	2.36	0.60	1.34	0.18	0.60
Submit	7.60	12.62	1.27	2.97	4.40	4.34	3.00	3.03
Bookmark	8.00	1.41	4.55	2.16	8.00	6.44	4.55	2.42
Change	43.60	26.38	2.45	5.63	28.40	19.96	3.36	3.11
Close	4.20	3.96	0.36	0.67	6.00	9.82	2.36	1.86

Figure 1 shows the average number of pages viewed for each task by the graduate and the undergraduate students, and Figure 2 shows the average viewing time per page for each task. First, we found that the graduates looked at significantly more non-results pages than results pages for both tasks in terms of the number of pages ($F(2,16) = 73.86, p < .01$). The undergraduates also looked at significantly more non-results pages than results pages for both tasks ($F(1,43) = 6.39, p < .05$; $F(1,43) = 107.82, p < .01$). The undergraduates looked at a significantly greater number of non-results pages particularly for the trip task ($F(1,43) = 43.39, p < .01$).

We found that the graduates showed no significant task-specific differences in the number of pages viewed during the search time. However, in the report task, the undergraduates spent significantly longer browsing non-results pages compared with the results pages ($F(1,43) = 7.60, p < .01$).

Analysis of Web-search Categories

We analyzed the number of search-related actions for the two tasks and two groups. We defined 10 categories of action to analyze user behavior on the Web. Table 1 lists the averages and standard deviations for the number of actions carried out for each task by the graduate and undergraduate students. The 2-factor analysis results revealed significant differences between the two groups for the Search, Next, Jump, and Browse actions. The undergraduates were significantly more likely to click links during the trip task than during the report task. The undergraduates tended to return to previous pages more often than the graduates, but the graduates tended to submit more forms than the undergraduates. The graduates bookmarked significantly more pages and switched to different tabs or windows significantly more often than the undergraduates. We also found that the graduates switched to different tabs or windows significantly more often for the report task than for the trip. The graduates also tended to close windows or tabs more often than the undergraduates. We noticed that both groups tended to close more windows and tabs in doing the trip task than during the report.

Summary of Behavioral Data Analysis

First, we consider task-specific differences in search behavior. We found that there were no task-specific differences commonly observed both groups. However, the two groups did share certain characteristics in the number of results pages they looked at and in their actions of Search, Next, Jump, and Browse. This suggests that both groups pur-

Figure 3: Location of blocks in *lookzone*

sued similar search procedures, particularly with respect to results pages, regardless of the type of task or level of experience.

After that, we considered the differences in search behavior that could be attributed to different levels of experience. In contrast to the graduates who tended to look at about the same number of pages for the same length of time in both tasks, we found that the undergraduates examined non-results pages for longer periods when doing the report task. Moreover, the search action data revealed that the graduates tended to change between windows and tabs and close them more frequently than the undergraduates. This reflected a tendency on the part of the graduates to search in parallel by frequently switching back and forth between a number of pages that were open at the same time. By contrast, the undergraduates were more likely to search sequentially by using the Link and Back functions to go back and forth between links.

3.2 Analysis of Eye-Movement Data

This subsection explains our analysis of the eye-movement data. Because Web searches involved dynamic changes in screen (scrolling and page transitions), no thorough assessment of search behavior could be based sorely up on quantitative analysis using stationary point coordinates. Tagging was also needed to determine exactly what the participants were looking at on the screen. We therefore employed a results page with a relatively simple structure in our investigations.

Definition of Lookzone

We defined 22 *lookzone* blocks on the page to classify exactly where participants were looking on the page. Figure 3 shows the 22 lookzone blocks superimposed on the Google-search results page. These same lookzone block items were applied to the search-engine pages used by the participants in this study.

Next, we captured images from the eye-tracking data of the participants at 0.5-second intervals, beginning as soon as the results pages were presented to them. We then manually tagged where the eye-gaze points in the extracted images fell within the lookzone. On the basis of this tagged data, we analyzed the number of eye-gaze points per block, and the eye-gaze points and number of clicks per search-result ranking.

Table 2: Average number of eye-gaze points for each Lookzone block

Lookzone	Report-writing task		Trip-planning task	
	Graduates	Undergraduates	Graduates	Undergraduates
1 Title bar	0.40	3.78	0.80	1.00
2 Menu	1.80	0.22	0.00	0.11
3 Bookmark	0.00	3.78	0.20	0.00
4 Tool bar	0.40	1.78	0.40	1.22
5 URL bar	0.40	0.78	0.00	0.11
6 Search bar	4.00	0.00	4.00	0.00
7 Search bar button	0.20	0.00	0.20	0.00
8 Tab	10.20	8.11	6.00	9.22
9 Link for services	2.40	17.67	2.20	5.00
10 Query box	5.40	36.89	3.00	12.56
11 Search button	0.00	0.89	0.20	0.67
12 Scroll bar	0.60	0.11	0.00	0.00
13 Number of hits	0.00	0.44	0.60	0.00
14 Sponsor link	0.00	6.67	11.40	12.11
15 Spell check	0.00	0.00	0.20	0.00
16 Title	38.80	60.67	39.20	42.11
17 Snippet	70.00	91.11	28.40	37.00
18 URL	16.60	40.89	12.40	15.44
19 Related search	1.20	3.00	1.20	2.56
20 Link for next page	1.00	0.78	1.00	0.78
21 Find in a page	0.00	0.00	0.00	0.00
22 Status bar	0.00	1.78	0.00	0.00
Out of lookzone	19.60	52.89	17.00	18.22
Lack of eye position	12.00	83.44	7.20	70.78

Analysis of Eye Gaze Points for Each Lookzone Block

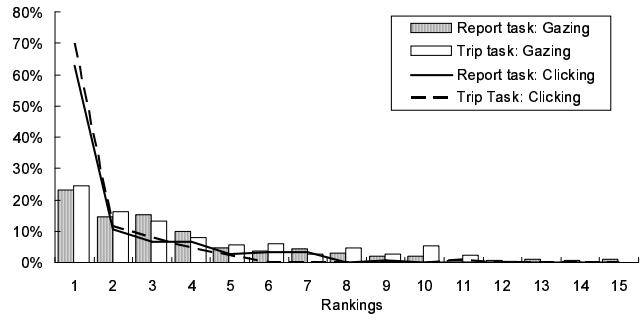
Table 2 shows the average number of eye gaze points for each Lookzone block broken down by task for the two groups. The category "Out of lookzone" is the number of eye-gaze points elsewhere on the page besides the 22 lookzone blocks, and the category "Lack of eye position" is the number of images in which the eye-gaze points could not be determined. As we can see from the table, most of the eye-gaze points on the results pages were focused on information pertaining to the hit pages (titles, snippets, and URLs).

The 2-factor analysis of variance results revealed clear differences between the two groups of students for a number of lookzone blocks. The undergraduates exhibited significantly more eye-gaze points on the tool bar ($F(1, 12) = 12.40$, $p < .01$). They also tended to focus more attention on the query box ($F(1, 12) = 3.87$, $p < .10$) and the search button ($F(1, 12) = 4.72$, $p < .10$). The graduates were significantly more prone to look at the search bar ($F(1, 10) = 6.02$, $p < .05$).

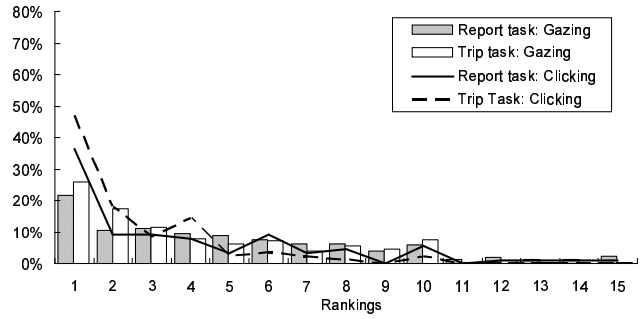
We also found differences between the tasks for a number of lookzone blocks. There were significantly more eye-gaze points on the scroll bars ($F(1, 12) = 4.77$, $p < .05$) and snippets ($F(1, 12) = 8.89$, $p < .05$) for the report task as opposed to the trip task. By contrast, in the trip task, students were more inclined to look at the sponsor's information ($F(1, 12) = 5.95$, $p < .05$).

Analysis of Eye-Gaze Points and Clicks for Each Ranking

As previously noted, there was a clear tendency for students to focus on the titles, snippets, and URLs of the hits displayed on the results pages. We consequently grouped the eye-gaze points on titles, snippets, and URLs and assigned rankings, then analyzed which rankings attracted the most views. We next extracted actual click-ranking data from the search-log data and investigated the relationship between eye-gaze points and clicks.



(a) Graduate students



(b) Undergraduate students

Figure 4: Percentages of clicks and eye-gaze points for each ranking

Figure 4 plots the percentages of clicks and eye fixations for all rankings for the two groups of students. One can see that the percentages are greatest for Rank 1 for both graduate and undergraduate students. These results are similar to those reported in previous studies [2]. After Carrying out a 2-factor analysis of variance on all ranks as a function of clicks, we found that the number differed significantly depending on the level of experience and type of task for Ranks 1, 6, 8, and 10. First, we found that the graduates tended to select rank 1 much more frequently than the undergraduates in doing the report task ($F(1,12) = 3.18, p < .10$). The graduates also selected rank 1 more often when doing the report task as opposed to the trip task ($F(1,12) = 5.68, p < .05$). Moreover, we found that both groups of students tended to select rank 6 more often for report tasks than for trip tasks($F(1,12) = 3.85, p < .10$). The undergraduates chose ranks 8 and 10 more often than the graduates did for both tasks (rank8:$F(1,12) = 5.36, p < .05$, rank10:$F(1,12) = 4.20, p < .10$). This reveals that the graduates tended to favorably assess and choose higher ranking pages from the search results, while the undergraduates tended to choose pages ranked 5 and below.

Next, we did a 2-factor analysis of variance on eye-gaze points for all rankings and found that the main effect of the task was quite significant in ranks 4 and 7 (rank4: $F(1,12) = 5.10, p < .05$, rank7: $F(1,12) = 6.12, p < .05$). This demonstrated that both graduates and undergraduates tended to examine lower ranking pages when conducting report tasks.

Summary of Analysis of Eye-Movement Data

We investigated to see if the eye-gaze points for each look-zone block in the eye-movement data, the eye-gaze points for each rank, and the number of clicks were correlated in any way with the different tasks and levels of experience. Analysis of the eye-movement data did reveal any task-specific differences. We found that the students in the report task perused from higher to lower ranking pages and scrutinized snippets revealing the content of the pages. By contrast, the participants had much less inclination to look at lower ranking pages for the trip task and focused more attention on the sponsors' information. This means that the type of task clearly did affect the information that was regarded as important and how students viewed the rankings. We found a clear tendency in graduates to look at the search bar at the top of the browser and to select more rank 1 pages. In contrast, the undergraduates tended to look more at the query boxes and search buttons at the top and bottom of the results page. Moreover, they were more likely to choose lower ranking pages. These characteristics observed in two groups suggests that the level of experience was clearly reflected in different search strategies and in the criteria for selecting ranked pages.

4. CONCLUSIONS

We studied how different tasks and levels of experience affect the behavior of students searching for information on the Web. Based on our analysis of search behaviors and eye-movement data, we found that the type of task and level of experience did indeed affect their search behaviors.

However, there were too few participants to allow reliable conclusions. In the future work, we will conduct more large-scale experiments to verify our findings.

5. REFERENCES

[1] A. Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, 2002.

[2] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. G. ay. Accurately interpreting clickthrough data as implicit feedback. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 154–161, New York, NY, USA, 2005. ACM.

[3] M. Kellar, C. Watters, and M. Shepherd. A field study characterizing web-based information-seeking tasks. *Journal of the American Society for Information Science and Technology*, 58(7):999–1018, 2007.

[4] G. Marchionini. Exploratory search: from finding to understanding. *Communications of the ACM*, 49(4):41–46, 2006.

[5] J. L. Moore, S. Erdelez, and W. He. The search experience variable in information behavior research. *Journal of the American Society for Information Science and Technology*, 58(10):1529–1546, 2007.

[6] A. Spink and B. J. Jansen. *Web Search: Public Searching of the Web*. Kluwer Academic Publishers, Dordrecht, the Netherlands, 2004.

[7] H. Terai, H. Saito, M. Takaku, Y. Egusa, M. Miwa, and N. Kando. Differences between informational and transactional tasks in information seeking on the web. In *Proceedings of the Second Symposium on Information Interaction in Context (IIiX 2008)*, pages 152–159, 2008.

[8] A. Thatcher. Web search strategies: The influence of web experience and task type. *Information Processing & Management*, 44(3):1308–1329, 2008.

Framework of a Real-Time Adaptive Hypermedia System *

Rui Li
Rochester Institute of Technology
102 Lomb Memorial Drive
Rochester, New York
14623-5608
rxl5604@rit.edu

Evelyn Rozanski
Rochester Institute of Technology
102 Lomb Memorial Drive
Rochester, New York
14623-5608
rozanski@it.rit.edu

Anne Haake
Rochester Institute of Technology
102 Lomb Memorial Drive
Rochester, New York
14623-5608
arh@it.rit.edu

ABSTRACT

In this paper, we describe a framework for the design and development of a real-time adaptive hypermedia system. The framework leverages on the integration of conventional adaptive hypermedia techniques and ACT-R architecture which serves as the theoretical background for the cognitive model that monitors the interaction process between users and the system. The users' information seeking skills in the hyperspace specified by their viewing patterns within the web pages and access patterns between the web pages are extracted from user tracing data. The user's viewing patterns are discovered by analyzing their fixation sequences with eyePatterns. The user's navigation strategies in the hyperspace are evaluated in terms of information foraging theory to serve as their access patterns. Both of these patterns are transformed into the knowledge stored in the cognitive model. Based on these interaction experiences between the user and the system, the cognitive model will re-arrange the presented information and the structure of the hyperspace in real time in order to facilitate the user to acquire valuable information as they perform information seeking tasks. Besides the flexible adaptability, this integration leads to the immediate feedback to assist the users' cognitive process to accomplish their information seeking tasks. The effectiveness of a conventional adaptive hypermedia system has been enhanced to a great extent.

Categories and Subject Descriptors

H.5.4 [**Information Interface and Presentation**]: Hypertext/Hypermedia—*architecture, navigation, user issues*; H.1 [**Information Systems**]: Models and Principles

General Terms

Design, Human Factors

Keywords

Adaptation, Cognitive Model, ACT-R Architecture, Information Foraging, Eye Tracking, Web Services

1. INTRODUCTION

Nowadays, both the amount and complexity of information are increasing exponentially, while the limited capability of information processing severely hampers humans to seek, gather and consume valuable information efficiently [10]. From this perspective, one of the most important studies in the information technology research field is how to maximize the allocation of human's attention to useful information rather than to simply provide people with access to the continuously-changing, chaotic, and overwhelming amount of information. Increasingly, massive amounts of information have been available to the average users in the form of hypermedia through the World Wide Web leading to the need for more adaptive and personalized websites. Adaptive hypermedia systems, as an alternative to the conventional "one-size-fits-all" websites [6], aim to augment web users' information processing capability. The basic idea of adaptive hypermedia systems is that by modeling individual user's particular goals, interests and preference, the system can tailor the content and format of the presented information to meet the user's special need in order to maximize their rate of gaining valuable information. Adaptive hypermedia systems can be widely adopted in many application fields, such as education [15] [13], e-commerce [9], and virtual environments [11]. The essential commonality is that users in these application fields have to explore reasonably large amounts of information with diverse goals and background knowledge.

The information structure of adaptive hypermedia systems consists of two interconnected spaces which are knowledge space and hyperspace. Knowledge space is a network model of the knowledge in a specific domain. The set of nodes in this structured domain model refer to a set of domain knowledge elements which can represent bigger or smaller pieces of domain knowledge depending on the particular application. The links among these nodes refer to their semantic relationships [4]. The hyperspace refers to the conventional web pages and page fragments connected by hyperlinks. The connections between these two spaces should be specified by the designers in order to assign web resources to the knowledge. As a crucial component, one of the most important functions of the domain model is to provide a framework to model users' domain knowledge and their goals. The majority of the adaptive hypermedia systems adopt overlay model

to simulate user's knowledge. The overlay model keeps a variable with each domain knowledge element to represent the estimation of user's knowledge level about this element. The user's goal is represented by a subset of domain knowledge elements to be learned. Currently, there is a trend in the research on adaptive hypermedia systems, especially in the online learning application field, tries to combine intelligent tutoring system with educational adaptive hypermedia by introducing "cognitive tutors" which are computational process models into adaptive hypermedia systems [5]. In the representative studies [15] [8], researchers integrated simple production systems with their adaptive hypermedia systems to guide the users' interaction with the systems. Besides the student model and goal model, these production systems can be considered as an adaptation model. Although just in its premature state, these adaptation models' effectiveness is relatively significant. This research trend partially inspired our study.

There are two major problems in the state-of-the-art adaptive hypermedia relating to user modeling and adaptation technologies. From the cognitive psychology point of view, the commercial platforms only provide a simple way of personalization and adaptation. Since the user model of the current adaptive hypermedia systems is no more than a record of a particular user's accumulative history visits, it fails to include some vital cognitive components that have a great effect on users' task performing process, particularly short-term memory, visual attention and misconception. As long as all these cognitive factors are treated appropriately, the adaptive hypermedia system can facilitate users' information seeking behaviors more efficiently.

Another problem hindering the further development of adaptive hypermedia systems is the lack of people with different expertise involved into the process of arranging personalized adaptive experiences to collaborate to achieve a good quality solution. Many adaptive hypermedia systems only serve as prototypes or research experiments without practical value. Consequently, how to integrate users' experiences into the hypermedia to direct the systems' adaptation and personalization is still a challenge. As the production systems were combined into the adaptive hypermedia systems, the system's effectiveness has been enhanced by providing real time feedback. However, these production systems are essentially committed to a particular use. This feature severely constrains the system's ability to acquire users' knowledge in a flexible form.

To solve these problems, we propose a computational process model built on ACT-R cognitive architecture as an embedded assistant to help users perform their tasks by adapting the hyperspace dynamically and providing real-time feedback. ACT-R cognitive architecture [2] aims to provide specification about human cognition. As an integration of various components of human cognition, ACT-R serves as a theoretical foundation to constructing cognitive models in order to produce coherent human behaviors in different environments. As the basic components of ACT-R architecture, the interaction between declarative knowledge and procedural knowledge enables our ACT-R model not only extends the conventional adaptation systems by providing a mechanism to acquire users' knowledge and skills in a more rapid and flexible manner, but also maintains trails of the users' information seeking process and cognitive states. These performances enable the adaptive hypermedia system to tailor the displayed contents and the structure of hyperspace in ways that improve the efficiency of the users' information seeking behaviors. Furthermore, information foraging theory [12] provides a new point of view to consider the interaction between users and adaptive hypermedia systems. According to this theory, humans can be viewed as informavores who actively seek, gather, and consume information in the culture environment in the same way as creatures like carnivores or herbivorous seek, gather, and consume food in the physical environment. In this sense, how to adapt the presented information to augment users' specific interests and needs can be converted into an optimization problem. We can evaluate and make sense of the user's information seeking behaviors by extracting their viewing patterns within web pages and navigation strategies between web pages in order to transform these patterns into the knowledge needed by the ACT-R model. It should be emphasized that compared to some previous attempts that focused on the learning field, our system aims at more general applications.

2. SYSTEM OVERVIEW

The overall structure of our real time dynamically adaptive hypermedia system is shown in Figure 1. In this structure, both the adaptive hyperspace and the domain knowledge space are components of the conventional adaptive hypermedia system. The user tracing model is responsible for observing and recording the interaction between users and the system to do further analysis. Eye tracking equipment and web logging software are used in this model to collect the eye movement data and the log record data respectively. Subsequently, the users' viewing patterns extracted from their fixation sequences within each web page will be analyzed by eyePatterns [16], and their access patterns are also compiled from the log records to serve as their navigation strategies between these web pages. ACT-R Model is used to learn and store the users' information seeking skills in order to direct the adaptation of the system to facilitate the users to acquire valuable information.

The user tracing model is responsible for evaluating the observed data from the users. These evaluations serve as the users' skills to be learned by ACT-R model in the form of declarative knowledge and procedural knowledge. Based on the observation data from user tracing model, the procedural rules stored in ACT-R model are activated to adapt the content and form of presented information dynamically to users' behaviors and provide necessary feedback in real time. The advantage of the ACT-R cognitive model is that it provides means of applying the psychological rules known from the users' cognitive behaviors to the adaptation of the interface, thereby improving the system's quality and usability.

Our system's adaptive behavior consists of two levels: adaptive presentation and adaptive navigation support. Adaptive presentation refers to adapting the content of a web page to the user's goals and knowledge background. In our system, the information fragments presented within a web page correspond to several Areas of Interest (AOI). Each of the AOIs contains a piece of information corresponding to the domain knowledge elements in the domain knowledge

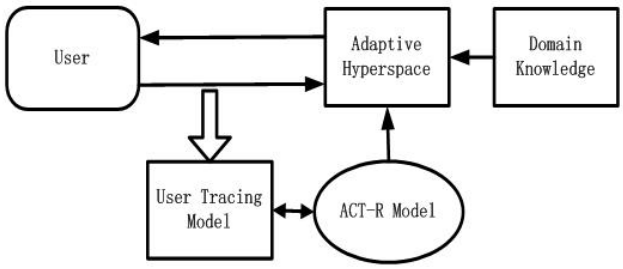


Figure 1: Overall System Structure.

space. The adaptive behaviors at the adaptive presentation level refer to hiding the AOIs which are assessed to be irrelevant to the tasks from the users and using an alternative way to present the displayed AOIs to emphasize their different priorities to the user's task. The adaptive navigation support is done in two ways: direct guidance which means the system highlights one of the links on the web page to indicate that this is the best link to follow, and web page sorting which means that the system sorts all the web pages according to the relevance evaluation stored as knowledge in our cognitive model: the more relevant the link is to the user's goal, the closer to the top in the hierarchical structure of hyperspace.

3. METHODOLOGY

3.1 User Tracing Model

The user tracing model consists of two operational modules: monitoring module and pattern extraction module. The monitoring module plays a role as a visual sensor to percept the users' interactive actions on the interface with eye tracking equipment and web log software. Pattern extraction module is capable of evaluating and recording the observed user's information seeking behaviors specified by two patterns which are the users' viewing patterns within a particular web page and the users' access patterns between the web pages.

The patterns will be mapped into the set of production rules which actively detect these inputs in ACT-R model. These rules update the declarative memory to contain chunks that represent the perceived behaviors which allow the system to adapt its displayed contents and structure of hyperspace. These observations of data enable the ACT-R model to adapt the information presented on the websites to users' cognitive process to pursue specific goals or interests as well as provide necessary instructions in real time to guide the users' navigation.

3.1.1 User Access Patterns

The users' access patterns between web pages refer to the users' navigation strategies in the website. These access patterns are specified by the data recorded in web server log. Since it records the user access behaviors of the website, web server log is still considered to be the most important source of data for the adaptive navigation support. Based on information foraging theory, we come up with a novel incremental optimization algorithm to evaluate the users' access patterns dynamically in order to enable ACT-R model to reorganize the structure of the website in real time. This algorithm not only identifies the set of web pages that are evaluated to be the most valuable to the user's task, but also provides criteria to re-organize the structure of hyperspace.

In [7], the log data shows that users spend shorter time on an index page choosing a link or topic and much longer time on a content page that they desire to read more thoroughly. We will extend this approach to distinguish index pages from content pages dynamically. In our model, the distinction between index pages and content pages is meaningful as long as it is defined for a specific user's navigation to perform a specific task in the website. Moreover, besides the time spent between content pages, the time spent within a content page is considered to evaluate the efficiency of the user access pattern.

According to information foraging theory, information presented in the culture environment is clustered into a set of patches, and each patch diffuse unique information scent. In our hierarchical structure hyperspace, the information displayed in each content page is viewed as one information patch. Consequently, the time taken by the users in a specific navigation to view the content page corresponds to the time spent within the patch, and the total time taken by the users in a specific navigation to get to the content page is defined as the time spent between information patches. Essentially, the time spent between content pages is defined as the sum of two parts. The first part is the time spent to choose links within index pages. The second part is defined as the total time taken by the user to download a series of web pages at different depths in the hierarchical structure of the hyperspace. Accordingly, the information scent for each of the links in the web page is specified by the activation level of the related chunks in the ACT-R model. Information diet refers to the user's selection of links to follow in order to gather valuable information efficiently [12]. The importance of information scent is that it is used by the users to assess the value of information gained per unit cost of processing the source. Based on these scent-based evaluations, the users are able to decide which links to follow so as to maximize the information diet. According to this, the rate of gain of valuable information per unit cost equals to the ratio of the total amount of valuable information that is necessary to be accessed for a particular task and the total amount of time cost within the content pages and between the content pages.

According to information diet model, the users assumed to be bounded rational always attempt to find relevant web pages in response to a goal or interest that are expected to contain most profitable information. The user's diet in the hyperspace refers to the rate-maximizing subset of the web pages that should be selected. The profitability of a content page is defined as the ratio of value gained from the content page to the cost of time within the page. Then the basic idea of our incremental optimized algorithm is that the users should continue to access content pages in the order of increasing rank of the pages' profitabilities as long as the profitability of the k+1 page is not less than the rate of gain for a diet of the top k pages. The algorithm outputs an optimized set of content web pages that should be accessed by the users to perform a specific task. The cumulative

gain function can be specified by the number of AOIs in the content web page and their mutual relevance with the user's goal which can be quantified by the spreading activation mechanism [1] in the declarative knowledge of the ACT-R model. The time spent in the process is recorded by web log software in the user tracing model. These optimized set of content web pages for a specific task will be transformed into declarative knowledge in ACT-R model.

3.1.2 User Viewing Patterns

The user's viewing patterns refer to the user's fixation sequences in the content pages and their efficiency of information seeking. According to [3], fixation sequence analysis can reveal the users' cognitive strategies to task completion that drive their attention to move around in the web page. A new tool used to discover the similarities in fixation sequences and identify the experimental variables that may affect their characteristics was described. This tool provides a solid practical foundation for constructing our user tracing model. Based on Yarbus' research work [17] that revealed that the order of fixations on regions of a stimulus is influenced by the relative importance of the regions to the viewer, and that viewers exhibited repeated cycles, or patterns, of fixations on the most interesting features of a stimulus, we assume that the users' eye fixations in a web page determine the efficiency of their information seeking behaviors in that page.

eyePatterns [16] will be adopted to extract the users' viewing patterns under a specific task in the pattern extraction module. A web page can be parsed into several sub-regions based on its layout and contents. These sub-regions are defined as area of interest (AOI) normally labeled with different characters. Therefore, the string representation of the fixation sequence corresponds to a concatenation of the AOI codes in the order of fixations occurred within the AOIs. eyePatterns is a software tool that provides several approach to discover the patterns in fixation sequences, moreover unknown and specified patterns can be found through discovery and pattern matching. these fixation sequences are integrated with the semantic meanings of each AOI [14].

3.2 ACT-R Model

To integrate cognitive models into the adaptive hypermedia system we need to keep track of the users' information seeking process and a series of cognitive states to adapt the layout and the structure of the hyperspace in a way to facilitate the effectiveness of information seeking. The key idea is that the cognitive model should incorporate the underlying information seeking skills that allow the users to pursue their goals or interests in an expected most efficient way. Based on the user tracing model, our system can monitor the users' information seeking behaviors and infer their intentions by mapping the behaviors to the components of the model. Subsequently, immediate adaptation of the hyperspace and real-time instructions can be generated to facilitate the users' information seeking behaviors.

3.2.1 ACT-R Architecture

The basic assumption in the ACT-R theory is that human cognition emerges through an interaction between a procedural memory and a declarative memory. Based on this,

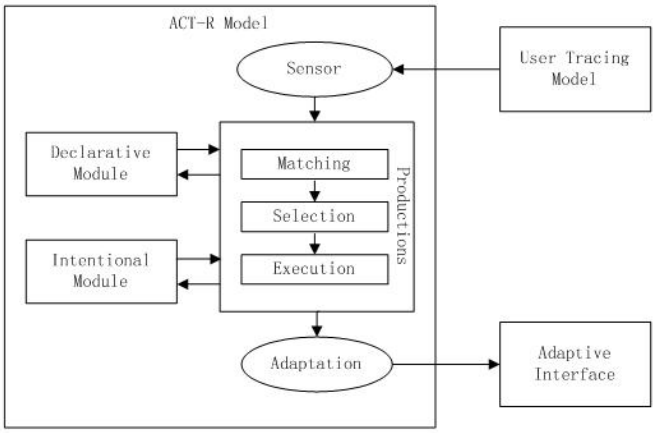


Figure 2: Architecture of the ACT-R Model.

there are several modules within the architecture of ACT-R. The declarative module retrieves information from long term memory, and the intentional module is used for keeping track of current goals and intentions of the users. A central production system is responsible for coordinating the behaviors of these modules. A production is a condition-action pair stored in the procedural memory. At a particular production cycle, once the condition parts of some productions are matched with the patterns from external world and internal modules, they will be gathered into the conflicting set. The conflict resolution will select only one production in each cycle to execute its action based on their utility. These actions make changes to the internal states of the modules and adaptive interface.

3.2.2 Declarative Knowledge

Declarative knowledge represents the various facts that people are aware they know and can explain them in an understandable way, such as the contents of a web page, the function of a certain button. Spreading activation mechanism is applied in the declarative memory to simulate the information retrieval process of human cognition. The declarative knowledge is grouped into a set of chunks, each of which contains a bigger or smaller piece of information depending on the applications. Parts of these pieces of information are corresponding to the contents displayed on the web pages of the hyperspace. An important feature of a chunk is its activation. The activation of a chunk represents to what extent this piece of information is needed at a particular time. The chunks connect to each other through associations which represent the co-occurrence between the pieces of information contained in the linked chunks. The associations have specific strengths to determine the amount of activation flow from one chunk to the related chunk. The users' goals or behaviors activate a group of chunks in this spreading activation network, meanwhile the contents displayed on the web page of the hyperspace activate some other chunks. These activations spreading via the associations through the network reflect the mutual relevance of the users' goals or behaviors and the contents displayed on the web page. All the associated chunks have been activated to a certain higher level.

3.2.3 Procedural Knowledge

The procedural knowledge which specifies how the declarative knowledge is transformed into active behaviors is represented by a set of production rules stored in the procedural memory system. These production rules detect activated declarative knowledge and input patterns from the sensor. At any point of time, multiple rules might be fired, but only one can be selected based on its utility to be executed. Since the user's information seeking behavior is specified by two kinds of patterns, the production rules should be designed accordingly such as:

IF *the current page is included in the optimized set of content web pages and it is a leaf,* THEN *it should be moved up in the hierarchical structure of the hyperspace and linked to an index page.*

IF *the AOI is with high relevance level in the current page and it is not included in the user's viewing pattern and the current page is included in the optimized set of content web pages,* THEN *the AOI should be highlighted.*

These are the English equivalent forms of the production rules that should be designed in our system.

4. REFERENCES

[1] J. R. Anderson. A spreading activation theory of memory. *Journal of Verbal Learning and Verbal Behavior*, 22(0):261–295, November 1983.

[2] J. R. Anderson. Act a simple theory of complex cognition. *American Psychologist*, 51(4):355–365, November 1996.

[3] J. R. Anderson and K. A. Gluck. *What Role do Cognitive Architecture Play in Intelligent Tutoring Systems?* Lawrence Erlbaum Associations, Philadelphia, PA, 2001.

[4] P. D. Bra and N. Stash. Aha! adaptive hypermedia for all. In *Proceedings of the AACE WebNet Conference*, pages 262–268. Association for the Advancement of Computing in Education, September 2001.

[5] P. Brusilovsky. *Adaptive Hypermedia: From Intelligent Tutoring Systems to Web-based Education*. Springer, Heidelberg, Berlin, 2001.

[6] P. Brusilovsky. *Authoring Tools for Advanced Technology Learning Environment*. Kluwer Academic Publishers, Dordrecht, 2003.

[7] G. Chibing and M. Nordahl. Building an adaptive website based on user access patterns. In *Proceedings of the 2005 International Conference on Cyberworlds*, pages 358–362. IEEE Computer Society, November 2005.

[8] F. Daniel, M. Matera, and G. Pozzi. Combining conceptual modeling and active rules for the design of adaptive web applications. In *Workshop Proceedings of the Sixth International Conference on Web Engineering*, pages 84–89. ACM, July 2006.

[9] A. Kobsa, J. Koenemann, and W. Pohl. Personalised hypermedia presentation techniques for improving online customer relationship. *The Knowledge Engineering Review*, 16(2):111–155, November 2001.

[10] G. Marchionini. Exploratory search: from finding to understanding. *Communication of the ACM*, 49(4):41–46, April 2006.

[11] J. Oberlander, M. O'Donnell, A. Knott, and C. Mellish. Conversation in the museum: Experiments in dynamic hypermedia with the intelligent labelling explorer. *New Review of Hypermedia and Multimedia*, 4(260):11–32, November 1998.

[12] P. Pirolli and S. K. Card. Information foraging. *Psychological Review*, 106(4):643–675, January 1999.

[13] C. Romero, S. Ventura, J. A. Delgado, and P. D. Bra. Personalized links recommendation based on data mining in adaptive educational hypermedia systems. In *Proceedings of the second European Conference on Technology Enhanced Learning*, pages 292–306. EC-TEL, March 2007.

[14] B. P. Stone and S. Dennis. Using lsa semantic fields to predict eye movement on web pages. In *Proceedings of the Twenty Ninth Conference of the Cognitive Science Society*, pages 665–670. Cognitive Science Society, August.

[15] W. Tarng, M.-Y. Chang, L.-K. Lai, S.-S. Tseng, and J.-F. Weng. An adaptive web-based learning system for the scientific concepts of water cycle in primary schools. In *the Sixth IASTED International Conference Proceedings*, pages 175–184. ACM, March 2007.

[16] J. M. West, A. R. Haake, E. P. Rozanski, and K. S. Karn. eyepatterns: Software for identifying patterns and similarities across fixation sequences. In *Proceedings of the 2006 Symposium on Eye Tracking Research and Applications*, pages 149–154. ACM, March 2006.

[17] A. L. Yarbus. *Eye Movements and Vision*. Plenum Press, New York, NY, 1967.

Inferring the Public Agenda from Implicit Query Data

Laura Granka
Stanford University
Google, Inc

granka@stanford.edu or
granka@google.com

ABSTRACT

Traditionally, implicit feedback measures are used to evaluate the performance of a particular information retrieval system. This research instead takes a distinctly applied approach to the use of implicit feedback, and extends the inference from aggregate query data to the social and political sciences. Using the three months prior to the 2008 election as a test scenario, the analysis here assesses daily fluctuations in search coverage of candidates and issues as predicted by the amount of news coverage, proximity to election day, and public opinion poll ratings of the candidates. Findings indicate that aggregate shifts in topical search queries may in fact be a useful, inexpensive indicator of political interest.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Search and Retrieval – *relevance feedback, search process;* J.4. [**Social and Behavioral Sciences**]: Sociology.

General Terms

Measurement, Economics, Human Factors.

1. INTRODUCTION

Information retrieval systems have frequently used measures of implicit feedback to evaluate both the performance of a retrieval system and infer searcher satisfaction [1, 2]. Implicit feedback refers to measures that are unobtrusively obtained from a user search session, such as clicks, queries, reading time, session length, and page scrolling. Implicit feedback has been used most frequently to infer result relevance based on user click behavior and reading time [3, 4], and has been validated with eyetracking [3]. To date, little research has applied implicit feedback to situations beyond the actual retrieval system. As usage of online search engines is only increasing [5], it is important to understand how implicit search behavior can be applied to other domains to understand broader user conditions. This research presents results from one such analysis, and discusses additional ways in which implicit feedback can benefit the political and social sciences.

2. AGGREGATE USER FEEDBACK

Much of the work done in the social sciences depends heavily on survey and experimental research. Both of these methods, while extremely desirable when controlling for individual-level variables (e.g., education, age, gender, political affiliation), are both costly and time-consuming to instrument and analyze. These measures are also susceptible to self-report or self-selection bias, particularly for questions assessing civic engagement or interest in public affairs [6, 7]. The ability to gauge aggregate changes in public opinion and issue awareness, in an immediate and inexpensive manner, is often the most desired alternative. One currently untapped tool for this is publicly available search query data. One specific platform, Google Insights for Search [8], offers users the ability to access daily changes in query volume for specific searches in a specific geography and time period. Existing research using this tool has shown how search volume is both indicative and predictive of external events, from flu outbreaks [9] to economic activity [10].

2.1 Search Queries and Topical Interest

Online search is an active medium, meaning that a user has to explicitly make the effort to acquire information about a given topic by manually typing in a query. Because of this, online searches queries may be a strong behavioral indicator of what issues and topics are at the top of a user's mind. This, coupled with the lack of self-report bias makes search queries an attractive way to implicitly measure fluctuations and changes in political issue interest over time.

Existing political and media research has tracked changes in issue interest over time, though as previously mentioned, through surveys or experiments. Research has repeatedly shown that public perceptions of issue importance are shaped by the amount of news coverage of that issue [11, 12]. In other words, the issues receiving the greatest news coverage are judged to be the most important issues. Our first step is to conduct a systematic evaluation of whether behavioral data obtained via search query volume is also consistent with the conclusions of agenda setting research. In other words, how do real world events and news coverage motivate political search traffic? The issues covered most prominently in the media are typically the issues that people judge to be most important; as such, we would expect to see these perceptions of importance reflected in a greater volume of online searches.

Overall, we hypothesize a strong level of convergence between search queries and news volume. The more interesting insights in our analysis will likely be the deviant cases – instances where the search query volume for a topic or issue exceeds what might be expected by its respective news coverage. Certain issues may be marked by extended periods of search activity, potentially revealing the topics that sustain audience interest enough to pursue additional information past the peak of news coverage.

SIGIR'09, July 19-23, 2009, Boston, USA.

3. METHODS

Standard surveys gauge public interest in political issues by first assessing issue awareness, and secondly, measuring perceived importance (via a rating scale). Search queries have the advantage of being able directly measure the first dimension – issue awareness. In order to perform a search, an individual already has to know about the topic or individual being queried for. While we don't know the level of detailed knowledge an individual may have about this issue, we do know that the individual knows about the topic and is making an effort to find out more about it.

Second, perceived importance is a bit trickier to measure through queries, but can still be done in a more indirect approach. The degree of importance attributed to a given issue can be inferred from overall aggregate changes in query volume for that given topic. Deviation from the norm query volume can be easily exemplified with seasonal examples – for instance, using Google trends, it is clear to see that in the United States, searches such as mittens or gingerbread increase in December. One would expect to see a similar phenomenon for political issues: query volume will reflect the rise and fall of public interest. In sum, overall changes in query volume, or sudden spikes in query volume, are two potential ways to assess how prominent or "important" an issue may be at any given time.

3.1 Data Collection

The data for this research was taken from the three months prior to the 2008 presidential election – the 92 days from August 1, 2008 to October 31, 2008. Overall news coverage was measured by counting instances of issues and political candidates being covered in transcripts from the three major US news networks (ABC, CBS, NBC). Transcripts were obtained from the Vanderbilt transcript database.

Coverage for each candidate and issue was obtained on a daily basis, to ascertain the changing volume of news coverage for every single day during this three-month period. As an additional step, the news coverage data was normalized according to the same normalization scheme as the search query data (as described below), so that when necessary, means could be compared between the two data sources.

Daily query volume data was downloaded from Google Insights for Search [8], which is publicly available online. The range of data collected was over the same three-month period, and limited to US websearch traffic. The purpose of this analysis is to determine the domestic effects of the US presidential campaign, so queries and news coverage were specifically chosen to represent the US market. The query volume does not reflect the actual number of queries that Google received; rather, the data is normalized according to the highest point in the data set, which receives a score of 100 (e.g., if there were 12 million searches for Obama on September 3, that day would receive a score of 100. If there were 6 million searches for Obama on August 1, that day would receive a score of 50). Other normalization factors are used to account for base increases in search traffic over time due to growth in the online population.

The query distributions for individual issues and candidates can then be compared with network news coverage of that issue or candidate. While the query means are not useful points of comparison between issues (each query resides in its own normalized set of data), the standard deviations may be useful, as they are representative of how regular or irregular searches are for a given term, such as whether certain terms are more severely punctuated with spikes in traffic.

The selected issues varied in degrees of their newsworthiness and sensationalism. As measures of "hard news," or substantive issues to the US, we tracked occurrences of the terms Iraq, War, Economy, Unemployment, Health Care, Taxes, and Education. To assess more sensationalist or "soft news" coverage, the terms Joe the Plumber, Tina Fey, and Saturday Night Live were analyzed. News coverage and query volume for each candidate's name – Obama, Biden, McCain, and Palin – were also obtained.

4. RESULTS

4.1 Election Proximity, News, and Search

For many campaign issues, the volume of news coverage significantly influenced subsequent search volume. Table 1 presents regression results using news volume and proximity to Election Day as predictors for search query volume. For most issues and candidates, there was a significant relationship between the issues covered in the news and the issues that people were most interested in searching for. However, for the topics War, Unemployment, and Health Care, proximity to the election was more influential than news coverage. In other words, searches for these terms increased as Election Day grew closer, irrespective of news coverage.

Table 1. Predicting Query Volume of Campaign Issues

Issue	Econ	War	Unemp	Taxes	Iraq	Hltcar	Educ
Intrcpt	16.44 (1.86)	51.61 (1.72)	48.43 (3.41)	48.64 (2.11)	49.39 (2.60)	47.79 (4.21)	60.30 (3.72)
News Trans	1.16** (0.11)	0.24* (0.10)	0.65 (0.79)	0.70** (0.25)	0.98** (0.23)	1.32# (0.66)	2.42** (0.57)
Electn Prox	0.19** (0.49)	0.43** (0.02)	0.20** (0.07)	0.24** (0.05)	0.40** (0.04)	0.31** (0.06)	0.14* (0.06)
St. Er. Reg	8.70	5.35	16.12	9.85	9.52	16.17	14.65
R^2 F Stat	0.81 189	0.82 200.3	0.13 6.47	0.48 40.82	0.59 64.48	0.25 14.8	0.23 13.48

Table 2. Query Volume for Candidates and Personalities

Issue	McCain	Obama	Palin	Joe the Plumber	Tina Fey
Intercept	-1.22 (46.39)	-74.40 (47.37)	5.09 (2.55)	-0.38 (1.38)	2.67 (3.60)
News Transcripts	0.78** (0.12)	0.45** (0.08)	0.87** (0.09)	2.97** (0.17)	4.59** (0.81)
Election Proximity	0.15** (0.05)	0.28** (0.08)	-0.01 (0.05)	0.02 (0.03)	0.18* (0.07)
Poll Data	0.09 (1.06)	1.69 (1.06)	—	—	6.40
St.Error Regression	11.72	10.09	—	—	17.13
R^2 F-Stat	0.50 29.81	0.66 57.91	0.57 57.94	0.82 197.4	0.38 27.2

Standard errors are reported in parentheses
No observations = 92
Significant p-values are indicated: ** p<.001, * p<.01, # p=.05

This may indicate that searches for these topics are driven by interest or perceived importance, potentially signaling that these issues are important to searchers. An October, 2008 Gallup report indicates that the key issues important to voters were the economy, gas prices, Iraq, healthcare, and terrorism [14]. While gas prices and terrorism were not included in this analysis, the results from this study did compare with the Gallup results, as searches for economy, Iraq and healthcare increased prior to the election (Table 1). Additionally, it was clear that broadcast news did not equally cover the issues of public concern. Figure 1 shows density plots of search queries and news coverage for two issues in our sample: economy and war. Economic news coverage fairly consistently predicts queries for economy; however, a similar trend does not exist when assessing news and queries for war.

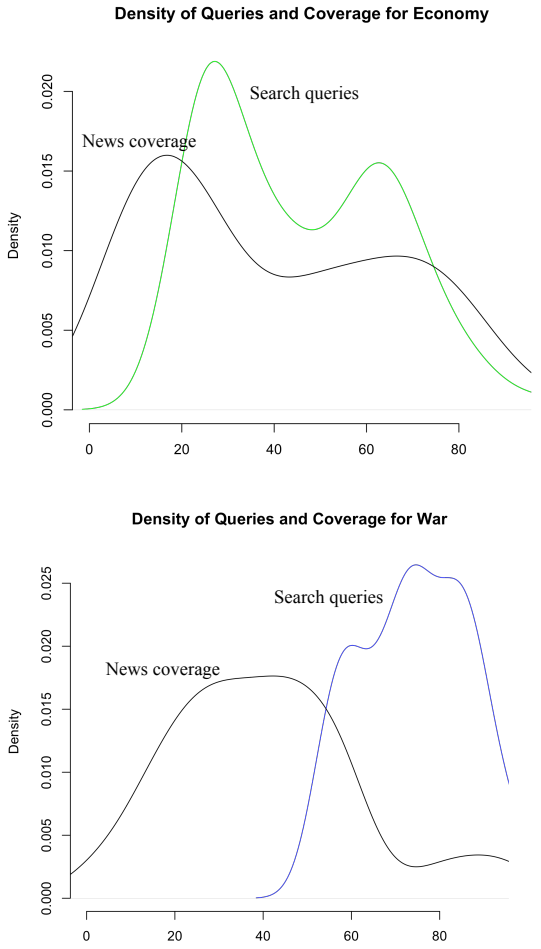


Figure 1. Top: density of news coverage (black) and search queries (green) for economy. Bottom: density of news coverage (black) and search queries (blue) for war. The x-axis represents the 92 days from Aug 1 – October 31, 2009.

4.2 Candidate Queries

Table 2 presents regression results predicting search queries for the candidates (Obama, McCain, Palin) and entertainment personalities (Tina Fey, Joe the Plumber), again using news volume and proximity to the election as predictors. A variable measuring public opinion approval, as assessed through polling data, was also included for the two presidential candidates. This data was obtained from Pollster.com, which aggregates multiple public opinion polls, and allows users to download data [13].

The regressions show that (as with issue searches), there is a signification relationship between the volume of news queries and the volume of searches for both candidates and entertainment personalities. As might be expected, the proximity to Election Day was only significantly influential for the two presidential candidates. The hypothesis that high approval in public opinion polls might influence search query volume was not supported – external measures of presidential approval (i.e., polls) do not appear to translate into increased search activity. This is particularly interesting, as it hints that political searches may be valence neutral; in other words, while it may be safe to say that queries measure interest, we cannot make the jump to conclude that greater search traffic also leads to support or approval.

Finally, in the final days leading up to the election, a number of searches increased. Searches for Obama spiked, as did searches for taxes. Prior to this, spikes in issue-based query traffic were limited to only one or two days, but immediately prior to the election, searches for these queries showed an increasing trend for multiple days. Recognizing how search volume changes directly before an electoral event could indicate the publics' attached importance to the particular issue.

4.3 Differences in News and Query Volume

Figures 2 and 3 present graphical differences between the news coverage and query volume of the presidential candidates and entertainment personalities. From these graphs, it is clear that search volume and news coverage are punctuated by key events in the campaign, such as political announcements and conventions. For some of these instances, particularly with individuals such as Sarah Palin and Joe the Plumber, who were previously unknown, the surge in query volume can also likely be attributed to novelty and curiosity – when a relative unknown comes on the scene, we may expect unsustainable spikes in query volume to learn about the newcomer.

It is also evident that news coverage of issues does not always generate equivalent spikes in search traffic, and furthermore, sometimes the spikes in query volume last longer than the increases in news coverage. Specifically, on October 16th (the day following Joe the Plumber's mention in the 3rd presidential debate), searches for Joe the Plumber surpassed online search activity for Obama and McCain, as people turned to the Internet to find out about this previously unknown individual.

To quantitatively compare the difference between news coverage and query volume for each candidate and entertainment persona, we conducted Welch two-sample t-tests between the normalized transcripts and normalized query volume. There was a comparable amount of news coverage and query volume for **Sarah Palin (transcripts = 17.21, queries = 17.36, t=-0.06, p =.955)** and **Joe the Plumber** (**transcripts =3.76, queries = 4.73, t=-0.37, p=.71**). The same was true for **Obama** (**transcripts = 35.32, queries 30.68, t=1.76, p=.08**).

There were significant differences between the amount of news coverage and the level of query volume for John McCain and Tina Fey. While **McCain** received significantly fewer online searches than what his news coverage might predict (**news = 40.31, queries = 28.06, t =4.55, p <.001**), **Tina Fey** generated significantly more online searches than what her news coverage might indicate (**news = 7.61, queries = 16.46, t=-3.19, p =.002**).

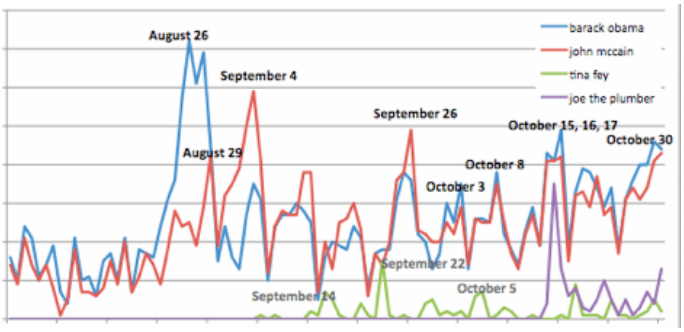


Figure 2. News transcripts for Obama (blue), McCain (red), Fey (green), Joe the Plumber (purple)

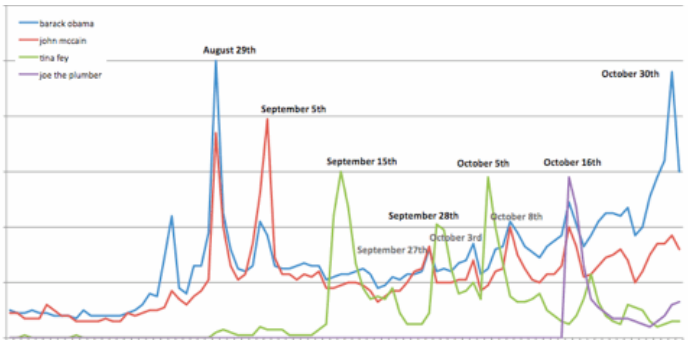


Figure 3. Search query volume for Obama (blue), McCain (red), Fey (green), Joe the Plumber (purple)

Why might this be? For the case of Tina Fey, it is likely that search activity surpassed news coverage because individuals wanted to watch (or re-watch) her SNL skits online. Searchers were not simply seeking out information, but additional media content in the form of videos and comedy clips from the show. McCain may have generated fewer queries than news coverage because he was already an established Senator (whereas Obama was largely an unknown), and individuals felt they needed to learn less about him.

5. FUTURE RESEARCH

The larger scope of this research effort is to take the first step at assessing how implicit feedback from the search process can effectively be applied towards the social sciences. The present study analyzed how fluctuations in query volume may be influenced by news coverage and external events. The degree of media influence on subsequent search activity is quite high, though in several cases (unemployment, war, healthcare), searches increased near Election Day irrespective of news coverage.

A logical next step is to gather real world data (e.g., unemployment claims/ layoffs, Dow Index/ interest rates) to compare changes in query volume with actual conditions. It will also be useful to gather public opinion data from National Election surveys to understand how search queries may fluctuate with survey data about issue importance.

To fully assess the impact of news, a more specific time-series analysis comparing news volume to changes in search query volume could be particularly informative: does media coverage always precede queries? What is the lag time before a news item becomes popularized in search volume? A multimodal analysis would also be interesting to more rigorously compare the spikes in query volume against the spikes in news coverage – for instance, what is it about some media events or news that causes query volume to increase much more than one would expect given the amount of news coverage. While this paper used network news transcripts as the predictor for news, future analyses may attempt to show whether different news sources, such as newspapers or web blogs, show stronger or weaker agenda setting effects.

Finally, the only form of implicit data used in this paper was aggregate query data. Subsequent analysis should also incorporate other typical measures of implicit feedback, such as reading time (to assess interest), clicks (from what sites did users acquire information), and query reformulation patterns. These additional measures, combined with a better understanding of how the voting electorate is represented in online search traffic will be useful in for making predictions about voter behavior or election results.

6. ACKNOWLEDGMENTS

Thank you to Shanto Iyengar, Solomon Messing, and Hilary Hutchinson who provided valuable feedback on earlier drafts of this paper.

7. REFERENCES

[1] Kelly, D. 2005. Implicit Feedback: Using Behavior to Infer Relevance. In eds, Spink, A & Cole, C. New directions in cognitive information retrieval, Springer: Netherlands.

[2] Fox, S., Karnawat, K., Mydland, M., Dumais, S., White, T. 2005. Evaluating Implicit Measures to Improve Web Search. ACM Transactions on Information Systems, 23, 2, 147-168.

[3] Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski, G. Gay. 2007. Evaluating the Accuracy of Implicit Feedback from Clicks and Query Reformulations in Web Search, ACM Transactions on Information Systems (TOIS), Vol. 25, No. 2.

[4] Radlinski, F., Kleinberg, R., Joachims, T. 2008. Learning Diverse Rankings with Multi-Armed Bandits. International Conference on Machine Learning, Helsinki, Finland.

[5] Fallows, Deborah. Search Engine Use. Pew. Aug 2008.

[6] Krosnick, J., 1999. Survey Research. Annual Review of Psychology, 50: 537-67.

[7] Hovland, C.I. 1959. Reconciling conflicting results derived from experimental and survey studies of attitude change. American Psychologist, 14, 8-17.

[8] Google Insights for Search. http://www.google.com/insights/search/

[9] Ginsberg, Mohebbi, Ginsberg, J., Mohebbi, M., Patel, R., Brammer, L., Smolinski, M., Brilliant, L. 2008. Detecting influenza epidemics using search engine query data. Nature.

[10] Choi, H. & Varian, H. 2009. Predicting the Present through Google search queries. April 2.

[11] McCombs and Shaw, 1972. The Agenda-Setting Function of Mass Media. Public Opinion Quarterly, 26, 176-187.

[12] Iyengar, S. & Kinder. 1984. News that Matters: Television and American Opinion. Chicago: U. of Chicago Press.

[13] Pollster.com http://pollster.com. Retrieved Dec 5, 2009.

[14] Newport, F. 2008. Obama has key edge on key election issues. Gallup Poll, June 24. Retrieved: http://www.gallup.com/poll/108331/Obama-Has-Edge-Key-Election-Issues.aspx

Evaluation of Digital Library Services Using Complementary Logs*

Maristella Agostii
University of Padua
Via Gradenigo 6/a, 35131
Padua, Italy
agosti@dei.unipd.it

Franco Crivellari
University of Padua
Via Gradenigo 6/a, 35131
Padua, Italy
crive@dei.unipd.it

Giorgio Maria Di Nunzio
University of Padua
Via Gradenigo 6/a, 35131
Padua, Italy
dinunzio@dei.unipd.it

ABSTRACT

In recent years, the importance of log analysis has grown, log data constitute a relevant aspect in the evaluation process of the quality of a digital library system. In this paper, we address the problem of log analysis for complex systems such as digital library systems, and how the analysis of search query logs or Web logs is not sufficient to study users and interpret their preferences. In fact the combination of implicitly and explicitly collected data improves understanding of behavior with respect to the understanding that can be gained by analyzing the sets of data separately.

Categories and Subject Descriptors

H.3.7 [**Digital Libraries**]: User Issues; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Search process*; H.3.4 [**Systems and Software**]: User profiles and alert services

General Terms

Algorithms, Design, Experimentation

Keywords

Web Log, Search Log, User Study

1. INTRODUCTION

The interaction between the user and an information access system can be analyzed and studied to gather user preferences and to "learn" what the user likes the most, and to use this information to personalize the presentation of results. User preferences can be learned explicitly, for example asking the user to fill-in questionnaires, or implicitly, by studying the actions of the user which are recorded in the search log of a system. The second choice is certainly less intrusive but requires more effort to reconstruct each search session a user made in order to learn his preferences.

SIGIR'09, July 19-23, 2009, Boston, USA.

Log is a concept commonly used in computer science; in fact, log data are collected by programs to make a permanent record of events during their usage. The log data can be used to study the usage of a specific application, and to better adapt it to the objectives the users were expecting to reach. In the context of the Web, the storage and the analysis of Web log files are mainly used to gain knowledge on the users and improve the services offered by a Web portal, without the need to bother the users with the explicit collection of information.

When research addresses the problem of studying log data in digital libraries, which are very complex systems, different characteristics regarding library automation systems and digital library systems need to be taken into account. In fact, for all the different categories of users of a digital library system, the quality of services and documents the digital library supplies are very important. Log data constitute a relevant aspect in the evaluation process of the quality of a digital library system and of the quality of interoperability of digital library services [2, 18]. With this concept in mind, it is also possible to think about new different logging formats which reflect how a generic DL system behaves [14].

This paper deals with the study of complementary types of logs in complex systems with the aim of finding new ways of using them to evaluate and personalize digital library services for the final users. The paper is organized as follows: Section 2 presents previous related work, Section 3 analyzes and presents different facets of the study and use of logs of complex systems, Section 4 presents the findings of the case study conducted in the context of the TELplus project[1] for the evaluation and personalization of the services of The European Library, and lastly Section 5 draws conclusions and indicates directions for the continuation of the work.

2. RELATED WORK

In the last decade, log analysis has become one of the main threads of research for understanding users of search engines as shown by the works presented at three major relevant conferences and that have been analyzed by us[2].

Those works study logs in different ways and for different

[1] http://www.theeuropeanlibrary.org/telplus/
[2] The three analyzed major conferences are:
SIGIR - http://www.sigir.org/
WWW - http://www.iw3c2.org/
JCDL - http://www.jcdl.org/

purposes, but they can be divided into two main classes: studies about search query logs, and studies about Web server logs. Since most of these research papers concern search engines, the focus of their research is more on improving queries and results and less on surfing the Web. The few exceptions to this classification will be analyzed later in the paper.

Query search logs can be used for: building knowledge, such as automatically building a search thesaurus [10], or acquiring ontological knowledge [24]; refining and expanding queries by means of analysis of search logs [4], or by means of correlations between query terms and document terms based on search query logs [11]; comparing of query extension techniques with pseudo-relevance feedback techniques [30]; organizing search results [29]; studying temporal changes and relationships, such as changes of queries on hourly basis in order to understand how user preferences change over time [5], analysis of multitasking user searches [6], issues related to ambiguity and freshness of queries [22], studies of causal relations between queries [27]; mining queries for extracting news-related queries [20], and association rules to discover related queries [25], or fast query recommendations [32].

Web logs can be used for: improving rank of results by replacing the adjacency matrix of the HITS algorithm with a link matrix which weights connections between nodes based on the usage data from Web server log traffic [21]; matching website organization with visitor expectations by means of Web log analysis [26]; finding user navigational patterns [9]; agents' detection [7].

There is also a recent emerging research activity about log analysis which tackles cross-lingual issues: [13] extends the notion of query suggestion to cross-lingual query suggestion studying search query logs; [16] leverages click-through data to extract query translation pairs. The interest in multilingual log analysis is also confirmed by initiatives promoted by the TrebleCLEF[3] coordination action which supports the development and consolidation of expertise in the multidisciplinary research area of multilingual information access (MLIA).

3. LOGS OF COMPLEX SYSTEMS

Present digital library systems are complex software systems, often based on a service-oriented architecture, able to manage complex and diversified collections of digital objects. One significant aspect that still relates present systems to the old ones is that the representation of the content of the digital objects that constitute the collection of interest is still done by professionals. This means that the management of metadata can still be based on the use of *authority control* rules in describing author, place names and other relevant catalogue data. A digital library system can exploit *authority data* that keep lists of preferred or accepted forms of names and all other relevant headings. This is a dramatic difference between digital library systems and search engines, and it is usually overcome with the analysis of log data. In fact a *search engine* often becomes a specific component of a digital library system, when the digital library system faces the management and search of digital objects by content in the same manner as information retrieval systems and search engines [1]. In all other types of searches, either the digital library system makes use of authority data to respond to final users in a more consistent and coherent way through a search system that is a sort of a new generation of online public access catalogue (OPAC) system, or the system supports the full content search with a service that gives the final users the facilities of a search engine.

Search query logs or Web logs alone give only a partial view of the stream of information that users produce. [28] show how to combine two different streams of data, search query logs and click-streams, in order to analyze re-finding behavior of a group of users under observation for a period of one year.

Moreover, log analysis can be supported and validated by user studies which are a valuable method for understanding user behavior in different situations. User studies require a significant amount of time and effort, so an accurate design of the process has to be carried out. In general, user studies and logs are used in a separate way, since they are adopted with different aims in mind. Ingwersen and Järvelin report in [17] that it seems more scientifically informative to combine logs together with observation in naturalistic settings. Pharo and Järvelin in [23] suggest systematic use of the triangulation of different data collection techniques as a general approach in order to get better knowledge of the Web information search process. An example of this type of combined studies is [15], where that authors claim that fully understanding user satisfaction and user intent requires a depth of data unavailable in search query logs but possible to acquire from other sources of data, such as one-on-one studies or instrumented panels.

The combination of implicitly and explicitly collected data improves understanding of behavior with respect to the understanding that can be gained by analyzing the sets of data separately. In particular for digital libraries, where the evaluation of the different services is difficult if logs are used alone, the combined sets of data provide the opportunity of reaching insights towards user personalization of digital library services.

From this starting point we have developed a method for collecting data derived from the user interaction log, "implicit" data, and data collected from user questionnaires, "explicit" data, for analyzing the interaction between users and digital libraries. This means that the conceived method is based on the combination and analysis of the following data sources: HTTP log which contains the HTTP requests sent by the Web client to the Web server during a user browsing session; search log which contains the actions performed by the user during a search; questionnaire data which are collected at the end of a user browsing and searching session.

The possibility of studying and correlating different sources of data was envisaged during the study of the Web portal of The European Library[4], which provides a vast virtual collection of material from all disciplines and offers interested visitors simple access to European cultural heritage.

[3] `http://www.trebleclef.eu/`

[4] `http://www.theeuropeanlibrary.org/`

4. RESULTS OF THE CASE STUDY

The European Library is a free service that offers access to the resources of 48 national libraries of Europe in 20 languages with about 150 million entries across Europe. The European Library provides a vast virtual collection of material from all disciplines and offers interested visitors simple access to European cultural heritage.

To validate the proposed method, a study was conducted in a controlled setting at the end of 2007 – beginning of 2008, in the computer laboratories of different faculties of the University of Padua, Italy, where students were requested to conduct a free navigation and search for information on The European Library portal and to fill in a questionnaire specifically designed to harvest the data that can be used to extract information on users satisfaction on the use of different parts of the portal. A total of 155 students participated in the study, mostly Italians, equally distributed between males and females, and with an age range typical of students of Bachelor and Master Degree (in most cases between 19 and 25 years old).

The analysis of the results was done in the following order: the analysis of each stream of data - i.e. HTTP log, search query log, questionnaires - was first conducted, while the analysis of possible interrelation among these sources was conducted later. The description of the analysis of each single stream is reported in [3], here we concentrate on the aspects which emerge from the correlation of the different sources of information.

Table 1 summarizes one of the important features when doing log analysis: session length. In particular, the table shows how different these lengths are according to the source that is analyzed. The "Search log" column shows the statistics of the times, in minutes, of sessions found in the search logs, and between brackets the times of sessions of users who registered to the portal. This shows that logging on is a clear intention of users who are willing to spend time in the portal and search more, compared to random users. The "HTTP log" column shows the times of sessions found in the HTTP logs computed in October 2007, and between brackets the times of the sessions of users who participated in the user study at the University of Padua. In this case, there is a strong bias of the students of the user study due to the time slot which was about 30/45 minutes. The times of random users are comparable to those found in the search logs. The last column shows the times of sessions for filling-in the questionnaires, which are obviously very similar to the times of HTTP sessions of the user study. There is one important aspect which emerges from the data: sessions are very short, browsing and searching activity lasts less than 2 minutes in 50% of the cases. This particular situation can be explained only by studying the answers of the users to the questionnaire where there are clear indications about some difficulties they found in understanding how to read the list of the results, and how to use some functions of the interface. These are also the reasons why they would have left the portal sooner if they had not been asked to stay and fill in the questionnaire.

An important interrelation was found among questionnaires and log data which may explain the short length of a user session. One of the outcomes of the questionnaire was the disorientation of the user upon entering The European Library portal for the first time, in particular it seems not to be clear what kind of information can be accessed through this portal. Users are in general ready to search in a Google-like fashion and obtain documents, in terms of links to pages or documents online, in the case of The European Library they are essentially in front of an online public access catalogue which retrieves bibliographic records. Obtaining library catalogue records after a search is a source of confusion which leaves the user unhappy and willing to leave the portal quickly.

Table 1: Summary of statistics for the time of a user session in minutes calculated in the search logs (between brackets registered user only), HTTP logs (between brackets user who participated in the study), and the time for filling-in the questionnaire.

	Search log	HTTP log	Questionnaire
Median	2.0 (4.0)	1.3 (30.25)	31.0
Mean	6.0 (8.0)	4.7 (31.80)	33.0

Questionnaires also show that images in particular seem to be very appealing for users; both the "treasures" section, a section which shows high resolution images of ancient documents, and the "exhibition" section, a section which shows pictures of the national libraries buildings, were thoroughly browsed by users even before making any query in the portal. This is an important clue which may suggest that there should be more linking from the images to the catalogue records. The interrelation among the information about users who prefer images and the HTTP log and searches log is still under investigation. In fact, we would like to see if this willingness expressed in the questionnaire is also reflected in user actions: for example, a user who is interested in images clicks more frequently on images or search for documents like maps or paintings; or a user expresses this interest in images but actually does not perform any action in the portal which confirms this interest.

5. CONCLUSIONS

The insights gained by analyzing log data together with data from controlled studies are more informative than the results that can be derived by separately analyzing the groups of data. Our studies on logs combined with interviews have shown that the results are more scientifically informative than those obtained when the two types of studies are conducted alone. This encouraging result constitutes the ground on which we are generalizing and formalizing starting from the obtained results. A crucial feature in the future will be making active use also of the information on metadata that are present in the log, because until now no active way of using them has been incorporated in the proposed method.

6. ACKNOWLEDGEMENTS

The work has been partially supported by the TELplus Targeted Project for digital libraries, as part of the eContent*plus* Program of the EC, and by the TrebleCLEF Coordination Action, as part of the 7FP of the EC.

7. REFERENCES

[1] M. Agosti, editor. *Information access through search engines and digital libraries*. Springer, Berlin, Germany, 2008.

[2] M. Agosti. Log data in digital libraries. In M. Agosti, F. Esposito, and C. Thanos, editors, *IRCDL*, pages 115–122. DELOS: an Association for Digital Libraries, 2008.

[3] M. Agosti, F. Crivellari, and G. M. Di Nunzio. A method for combining and analyzing implicit interaction data and explicit preferences of users. Workshop on Contextual Information Access, Seeking and Retrieval Evaluation (ECIR 2009), April 2009.

[4] P. G. Anick. Using terminological feedback for web search refinement: a log-based study. In *SIGIR*, pages 88–95. ACM, 2003.

[5] S. M. Beitzel, E. C. Jensen, A. Chowdhury, D. A. Grossman, and O. Frieder. Hourly analysis of a very large topically categorized web query log. In M. Sanderson, K. Järvelin, J. Allan, and P. Bruza, editors, *SIGIR*, pages 321–328. ACM, 2004.

[6] N. Buzikashvili. An exploratory web log study of multitasking. In Efthimiadis et al. [12], pages 623–624.

[7] N. Buzikashvili. Sliding window technique for the web log analysis. In Williamson et al. [31], pages 1213–1214.

[8] L. Carr, D. D. Roure, A. Iyengar, C. A. Goble, and M. Dahlin, editors. *Proceedings of the 15th international conference on World Wide Web, WWW 2006, Edinburgh, Scotland, UK, May 23-26, 2006*. ACM, 2006.

[9] J. Chen and T. Cook. Mining contiguous sequential patterns from web logs. In Williamson et al. [31], pages 1177–1178.

[10] S.-L. Chuang, H.-T. Pu, W.-H. Lu, and L.-F. Chien. Auto-construction of a live thesaurus from search term logs for interactive web search. In *SIGIR*, pages 334–336, 2000.

[11] H. Cui, J.-R. Wen, J.-Y. Nie, and W.-Y. Ma. Probabilistic query expansion using query logs. In *WWW 2002*, pages 325–332, 2002.

[12] E. N. Efthimiadis, S. T. Dumais, D. Hawking, and K. Järvelin, editors. *SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, USA, 2006*. ACM, 2006.

[13] W. Gao, C. Niu, J.-Y. Nie, M. Zhou, J. Hu, K.-F. Wong, and H.-W. Hon. Cross-lingual query suggestion using query logs of different languages. In Kraaij et al. [19], pages 463–470.

[14] M. A. Gonçalves, G. Panchanathan, U. Ravindranathan, A. Krowne, E. A. Fox, F. Jagodzinski, and L. N. Cassel. The xml log standard for digital libraries: Analysis, evolution, and deployment. In *JCDL*, pages 312–314. IEEE Computer Society, 2003.

[15] C. Grimes, D. Tang, and D. M. Russell. Query logs alone are not enough. In E. Amitay and C. G. M. J. Teevan, editors, *Query Log Analysis: Social And Technological Challenges. A workshop at the 16th International World Wide Web Conference (WWW 2007)*, May 2007.

[16] R. Hu, W. Chen, P. Bai, Y. Lu, Z. Chen, and Q. Yang. Web query translation via web log mining. In S.-H. Myaeng, D. W. Oard, F. Sebastiani, T.-S. Chua, and M.-K. Leong, editors, *SIGIR*, pages 749–750. ACM, 2008.

[17] P. Ingwersen and K. Järvelin. *The Turn*. Springer, The Netherlands, 2005.

[18] T. Koch, A. Ardö, and K. Golub. Browsing and searching behavior in the renardus web service a study based on log analysis. In H. Chen, H. D. Wactlar, C. chih Chen, E.-P. Lim, and M. G. Christel, editors, *JCDL*, page 378. ACM, 2004.

[19] W. Kraaij, A. P. de Vries, C. L. A. Clarke, N. Fuhr, and N. Kando, editors. *SIGIR 2007: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Amsterdam, The Netherlands, July 23-27, 2007*. ACM, 2007.

[20] M. Maslov, A. Golovko, I. Segalovich, and P. Braslavski. Extracting news-related queries from web query log. In Carr et al. [8], pages 931–932.

[21] J. C. Miller, G. Rae, and F. Schaefer. Modifications of kleinberg's hits algorithm using matrix exponentiation and weblog records. In W. B. Croft, D. J. Harper, D. H. Kraft, and J. Zobel, editors, *SIGIR*, pages 444–445. ACM, 2001.

[22] J. Parikh and S. Kapur. Unity: relevance feedback using user query logs. In Efthimiadis et al. [12], pages 689–690.

[23] N. Pharo and K. Järvelin. The SST method: a tool for analysing Web information search processes. *Information Processing & Management*, 40(4):633–654, July 2004.

[24] S. Sekine and H. Suzuki. Acquiring ontological knowledge from query logs. In Williamson et al. [31], pages 1223–1224.

[25] X. Shi and C. C. Yang. Mining related queries from search engine query logs. In Carr et al. [8], pages 943–944.

[26] R. Srikant and Y. Yang. Mining web logs to improve website organization. In *WWW 2001*, pages 430–437, 2001.

[27] Y. Sun, K. Xie, N. Liu, S. Yan, B. Zhang, and Z. Chen. Causal relation of queries from temporal logs. In Williamson et al. [31], pages 1141–1142.

[28] J. Teevan, E. Adar, R. Jones, and M. A. S. Potts. Information re-retrieval: repeat queries in yahoo's logs. In Kraaij et al. [19], pages 151–158.

[29] X. Wang and C. Zhai. Learn from web search logs to organize search results. In Kraaij et al. [19], pages 87–94.

[30] R. W. White, C. L. A. Clarke, and S. Cucerzan. Comparing query logs and pseudo-relevance feedbackfor web-search query refinement. In Kraaij et al. [19], pages 831–832.

[31] C. L. Williamson, M. E. Zurko, P. F. Patel-Schneider, and P. J. Shenoy, editors. *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, 2007*. ACM, 2007.

[32] Z. Zhang and O. Nasraoui. Mining search engine query logs for query recommendation. In Carr et al. [8], pages 1039–1040.

Watching Through the Web: Building Personal Activity and Context-Aware Interfaces using Web Activity Streams

Max Van Kleek
MIT CSAIL
32 Vassar Street
Cambridge, MA 02139
USA
+1-617-669-3864
emax@csail.mit.edu

David R. Karger
MIT CSAIL
32 Vassar Street
Cambridge, MA 02139
USA
+1 (617) 258-6167
karger@csail.mit.edu

mc schraefel
School of Electronics and
Computer Science
University of Southampton
SO17 1BJ, United Kingdom
mc+sigir@ecs.soton.ac.uk

ABSTRACT
This paper proposes the use of the increasing numbers of Web-based user activity and personal information sources to enable the creation of more personal, adaptive, and activity-sensitive information tools. We describe our initial steps at investigating this idea, including challenges surrounding integrating information from heterogeneous web data sources. This paper contributes an implementation of an in-browser framework called PRUNE that derives an internal world model consisting of an entity database and event chronology based on heterogeneous RSS/ATOM feeds, Web APIs and other web-based data sources. Finally, we apply this model in an application called Notes that Float, that automatically learns associations between notes and a user's other activities to enable context-aware implicit reminding.

Keywords
User modeling, life-logging, personalization, and personal information management

1. Introduction
The wealth of instantaneous information brought to us by the Web, e-mail, mobile phones, social networking web sites and ubiquitous network access has begun to dramatically change how we manage our everyday work and leisure activities. In particular, the sheer volume of information has exceeded our ability to consume it, while at the same time our new responsibilities demand that we stay on top of it -- to keep abreast of the status of our family, friends, colleagues, field, economic conditions, financial market, and so on. These heightened demands on our ability to process, find, and filter information prescribe the need for better personal information tools that expand our ability to pay attention to, and act upon, the vast quantity of information arriving for us and that we have collected in our personal information repositories.

Our goal, in our research, has been to apply personal information to the management of personal information itself; specifically, to design personal information management tools that when supplied with information pertaining to its user's ongoing activities, tasks, situations and preferences, can proactively take appropriate action on the user's behalf.

In the remaining sections of this paper, we describe a framework for longitudinal activity monitoring using the web, and a simple prototype personal information management tool that uses a model derived from activity logs to enable context and activity-sensitive reminding.

SIGIR'09, July 19-23, 2009, Boston, USA.

2. User activity monitoring using the Web
This excitement surrounding social sharing on “Web 2.0” has stimulated the growth of an immense number and variety of "life-tracking" web sites that are making the chronicling everyday life activities into a popular pastime. Several of these sites have created applications to enable the automatic capture and publishing of activity data sensed via the user's own personal devices, such as their laptop, desktop, or mobile phone. Examples include Google Latitude[1], which senses the user's location using Wi-Fi, GPS and cell phone towers, Rescue Time[2], Slife[3] and Wakoopa[4], which track a users' application usage, and the audioscrobbler[5] from Last.fm, which tracks a user's music listening activity. Other sites such as fitbit[6] and Nike, sell hardware devices that capture and publish user activity to their respective sites, letting users visualize and track various metrics. The result of the introduction of these sites and their accompanying data capture tools is that hundreds of thousands of individuals have started broadcasting minute-by-minute updates of their daily life activities to the web. While the primary intended use of these data is for letting people compare their lives with others, most of these services offer the data back to users via Web APIs and syndication feeds (RSS/ATOM), turning these services into potential sources of data for adaptive and context aware-enabled applications.

Compared to directly sensing user activity, there are a number of drawbacks to using third party life-tracking sites. First, the fidelity and accuracy of user activity data acquired from the web is often lower, and is made available with substantially higher latency than if directly captured. In fact, we have witnessed a number of the sources seemingly deliberately degrading the quality of the data returned by their APIs such as by omitting certain properties or throttling query/update rates. Last.fm, for example, omits the "end time" of a played song, thus making it impossible to know the duration that the individual listened to a particular track. Furthermore, the very fact that such volumes of high-fidelity

[1] See http://latitude.google.com
[2] See http://www.rescuetime.com
[3] See http://www.slifelabs.com
[4] See http://www.wakoopa.com
[5] See http://www.audioscrobbler.net, associated with http://last.fm
[6] See http://www.fitbit.com

personal activity information are being automatically transmitted to random web services (where they are aggregated and kept indefinitely) should signal potential privacy concerns.

However, despite these disadvantages, we believe that the Web is a convenient source of a tremendous quantity of rich data about users that would have otherwise been able to obtain. For example, data aggregated from mobile phones, such as the user's call history, and a user's text messages sent and received, can easily be obtained via SkyDeck.com. Similarly, an individual's spending history, broken down by time of day and merchant, is available via Mint.com. Soon, each individual's health and medical history will be readily available via services such as Google Health. Furthermore, as these services were designed to facilitate sharing of this information with others, incorporating and obtaining information about friends' activities becomes straightforward. As the number and variety of applications that use data provided by these sites increases, we believe that these sites will be pressured to improve the quality of the data they make available via their APIs.

2.1 Modeling from heterogeneous data

While the web makes accessing the data itself convenient, building personalized applications using this data, particularly from multiple sites or sources requires addressing several challenges. First, despite standardized serialization formats (such as RSS/ATOM feeds, REST/JSON APIs), web sites typically publish data using schemas of dissimilar structure. For example, audioscrobbler RSS feeds have song and artist fields merged into a single field called "Title", while most other music-related APIs separate these out. Thus, in order for data from heterogeneous sources on the web to be effectively compared and combined, these differences and inconsistencies need to be dealt with.

Since it is undesirable to have to deal with the complexities of individual sources at the application level, we built a lightweight integration framework (called PRUNE[7]) to specifically handle this integration process. Based on data retrieved from external sources, PRUNE derives a simple world model that applications can query and explore directly. Having an intermediate model collapses the problem of schema alignment from an $O(n^2)$ pairwise alignment problem to an $O(n)$ alignment -- between external schemas and PRUNE's world model.

PRUNE's world model consists of two databases containing entities and events, respectively. Entities represent people, places, documents, events, and other "things" represented by various web data sources. Information about person entities are currently obtained from open social networking sites or web-based PIM tools such as Gmail contacts. Similarly, information about events can be acquired directly from a localizer, a gazetteer service, or event descriptions (which contain location descriptors). Relations between entities are represented by named properties on these entities. Events, on the other hand consist of time-based observations of the dynamic states or activities of those entities. Events are 5-tuples (start and time, event type, entity and state/value) representing the duration that the particular entity engaged in or assumed the particular value. Events are kept in an ordered chronology, which allows applications to easily examine sequences of events for building temporal models and analyzing correlations between states and activities.

New data sources can be added to enhance PRUNE's model. If the new data source uses the same schema as another site or source PRUNE uses already, it will be able to use data from the site directly. If not, the user may have to build an import filter, a short piece of Javascript that maps incoming fields to create/update operations on entities or events in the model. A tutorial on building such import filters makes it easy for novice programmers to construct such filters, and filters can be easily published and made available for use by other users.

With respect to predictive modeling, PRUNE's current modeling mechanisms are rudimentary, consisting of learning probabilities over event type states, and entity identity resolution. For the former, PRUNE supports online or batch learning of either full discrete probability distributions of events, or simple pair-wise co-occurrences (which can be used for Naive-Bayes style inference). These probabilities are either learned from event counts or event time durations corresponding to how long a user or entity assumes a given state. With respect to entity reference resolution, PRUNE assumes that every entity (such as a person, place or resource) at least one inverse functional property (which can be used as a unique key for merging data about entities from heterogeneous sources), and at least one familiar name. Familiar names may not necessarily be unique, and thus can only be used to retrieve entities, not modify them. This facility is used to identify mentions of people, places and things in interactions with users.

3. Notes that Float: Anticipating information needs using heterogeneous activity models

While the recent rise in popularity of personal, lightweight note-taking and scrap-booking tools have improved many individual's note capture frequency and volume, the abundance of the resulting notes can make effectively using and accessing particular notes difficult: in order for a particular note to be useful, the user must remember they took it (to make the effort of looking for it), or s/he must serendipitously rediscover it in her collection. As one's note collection grows, the likelihood of forgetting increases, while the likelihood of serendipitous discovery diminishes due to decreased visibility.

To address this problem, we have designed a system called "Notes that Float" (NTF) that proactively anticipates when a note might be needed based on its contents and previous access patterns. When NTF detects that a note might be useful in a particular new situation, it actively raises its visual salience by popping the note to the top of the user's list of notes. NTF was built on top of List-it [8], our simple personal note-taking tool for Firefox, and relies on PRUNE for observations of user activity.

3.1 Note content features (Dates and times)

Although we are currently expanding NTF to analyze other content features (particularly entity references and note types), we started with extracting date and time expressions for two reasons. First, they appeared prominently in a significant number of notes of our pre-study [1]. Second, these times often indicated when a particular event occurred or task to be done was due, and thus served as a useful indicator of times of future relevance. We designed NTF's date-time extractor to a wide variety of ways of referring to time, including vague and relative descriptions, and constructed NTF's expression relevance function to represent how likely it was that a particular expression referred to a particular

[7] PRUNE: PLUM Runtime Usually Not Exponential (PLUM = Personal Lifetime User Modeling, a previous, RDF-based life-tracking project, please see http://plum.csail.mit.edu)

moment in time. For example, the expression "tomorrow" yields a high likelihood of relevance for any calendar times that falls within the next (wall clock) day after the expression was written. Although this function was hand-constructed, we are working to replace it with one derived from a corpus such as TimeEx [7].

3.2 Correlating note use with activity/location

Our pre-study suggested that individual notes tended to be edited at particular times of day, and days of week, and while the user was looking at the same web pages as when the note was edited previously. NTF is designed to identify and leverage correlations (when present) between note-edits and any user activity or state, including time of day, physical location, weather, web page views, music listening activity or ongoing calendar activities, and use this towards ranking notes by relevance.

NTF's algorithm is simple: it listens to new events representing observations of changes to entities and their activities. These events might consist of observations of a change in what web page or document the user is viewing, the room in which they are sitting, or music to which they are listening. Then, whenever a note is accessed, NTF tallies a count, for each activity and situation dimension, of the particular activities, documents, locations, or other entities were being performed, viewed, or experienced at that particular time. These counts are then used directly in the ranking process, described next.

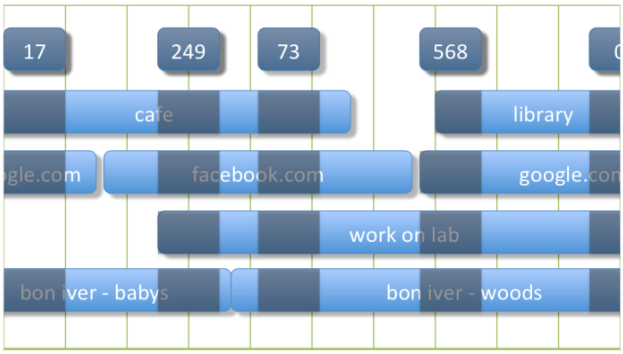


Figure 1. Learning note-activity relevance – To compute the relevance of a particular note (top row) to a particular activity or context (location, web site, scheduled calendar event or music listening activity), overlap counts are computed between the event and the other ongoing events at the same time (here, time is illustrated as flowing along the x-axis). Extremely brief overlaps are discounted.

3.3 Ranking notes

The learned associations allow NTF to simply rank notes by the posterior likelihood of the note given the user's active context and included date/time expressions. Specifically, the posterior relevance of each note is first calculated as follows:

$$\begin{aligned} &Relevance(note_i \mid note\ contents, user\ context) \\ &= P(note_i \mid Tr(note_i, now), C_1, C_2, \ldots C_{|C|}) \\ &\propto Tr(note_i, now) P(note_i) P(C_1, C_2, \ldots C_{|C|} \mid note_i) \\ &\propto Tr(note_i, now) P(note_i) \Pi_C P(C_c \mid note_i) \end{aligned}$$

Where $P(\text{note}_i)$ is used as shorthand to represent the prior probability that Note i is accessed, $Tr(\text{note}_i, \text{now})$ is the maximum time relevance (computed by NTF's time expression evaluation function) of all time expressions extracted from the note, and each $P(C_c|\text{note}_i)$ term in the final expression represents the probability that context dimension/activity type (e.g., "web page viewed") assumed a particular value (e.g., "http://mit.edu") while a note was being accessed. This value is directly computed from the pair-wise counts previously described by taking the ratio of counts for the particular value (e.g. viewing of "http://mit.edu" while accessing a note) and the sum of the counts of all values for that activity type (e.g., viewing *any* web page while accessing the particular note). In the third line above, we made a conditional independence assumption of each context type given a particular note. While this is an obvious simplification of actual fact, this is done to let the system use pair-wise affinities instead of full conditional probability tables (CPTs) which are space-inefficient and expensive to marginalize, and forces NTF to fit a simpler model corresponding to a Naive-Bayes independence assumption.

As described in the next section, the NTF UI allows users to select which event/activity types (C_c's) are included in the calculation above, as well as whether $Tr(\text{note}_i, \text{now})$ is included. This lets the user have more control over the ranking process.

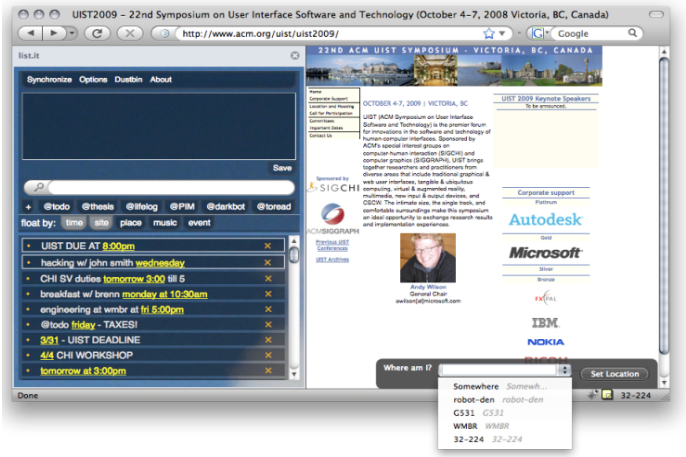


Figure 2. List-it interface – Sidebar on the left, with re-search bar, float by: bar, and notes with time information highlighted. On the bottom right shows the user's computed location.

3.4 User interface

Figure 2 shows the List-it note-taking tool embedded in the Firefox sidebar with the NTF extension installed. NTF introduces the small "float by" bar beneath the search tabs on the main UI, which is used to select floatation modes. Multiple modes may be enabled simultaneously, resulting in these terms being included as "givens" to the ranking algorithm previously described. When any of these buttons are enabled, NTF re-ranks all notes in List-it every 30 seconds (adjustable), bringing notes that exceed a relevance threshold to the top of the list. To make these notes salient and to differentiate them from the user's other notes, it "glows" floated notes with a white perimeter. When time-expression ranking is enabled, detected date/time expressions are also made to glow in yellow when the user mouses over them. The intention is to give the user feedback about the clues the system has used to rank the particular note in question. An additional configuration page (not shown) allows the user to configure PRUNE's data sources, including specifying their site-specific account usernames and passwords. Some data sources, such as our OIL localizer, require the user's system to have a WiFi card installed and, to "instruct" the system for training. Users can teach OIL about places (such as the rooms in their house) by

clicking on a small widget in their status bar, and either typing a new place or selecting one they previously selected. This creates a new location state and assigns the current Wi-Fi signature to it, so that it may be recognized on subsequent visits.

3.5 INITIAL EVALUATION

Ten existing List-it users volunteered to test an early alpha release of the NTF-enabled version of List-it for 5 days, in which only 3 floating modes were available: By Time, By Place (physical location) and By Site (website). Nine users successfully installed the system (one user could not due to an unforeseen compatibility issue with 64-bit Windows). Participants used By Time mode the most (26% of the time), followed by no ranking (24%), by Site alone (14%), and by Place alone (12%). Combined modes were less popular. During the study duration, NTF re-ranked notes a total of 73 times (across all users), recommending up to 10 notes per rank. We are planning a formal study and larger deployment after implementing a few features to enhance the usability and predictability of the system, as described next.

3.6 Ongoing NTF Work

The NTF work just described demonstrates our first steps at applying PRUNE to facilitate implicit contextual retrieval for personal note collections. Implicit contextual retrieval, we believe is important in the future for helping individuals manage large quantities of personal information, some of which they may have entirely forgotten about. Our initial trial, while small, ended with encouraging results; one participant said: "[Having] tried it I decided that I liked it .. This could be the answer to an older man's increasing info and fading memory problems."

With respect to next steps, we are working to improve the NTF ranking algorithm and UI in several ways. The NTF ranking algorithm was our naive initial first shot at devising a method that was simultaneously principled and could take into account heterogeneous activity, situational and content features of notes. One initial improvement will be to automate the selection of context types/dimensions used in the ranking process; this might simultaneously improve ranking performance and permit the simplification of the UI to a single button ("ranking on/off"). To do this, NTF could learn (e.g., using feature selection approaches) the dimensions of context that are most strongly correlated with use of particular notes. A note containing the username/password for a web site, for example, is likely to be correlated only with web site viewing activity but not others. Second, to measure the effectiveness of the ranking, we plan to add facilities that let users easily give feedback about floated notes in various ways. This feedback will allow users to express nuances of "I don't want to see this now" – differentiating, whether the recommendation was a bad one (so that this feedback may be used to adjust the particular notes associations), or whether the user wants to dismiss the reminder until later for other reasons – such as in the case of deliberately putting off a to-do item. Finally, we also want to allow for greater transparency of learned associations, so that users will be able to understand why particular notes were chosen and promoted by the algorithm.

4. Conclusion

In this paper we have described our initial work towards using "Web 2.0" user activity information sources to observe user activity and information access over time, and to apply this to the construction of an implicit information reminding service. Although in its early stages of development, our simple application, NTF, supports a level of flexible, implicit context and activity-sensitive predictive reminding not available in PIM applications today. Achieving this context-adaptivity would have been substantially more difficult to implement and maintain if we had written the low-level sensing and instrumentation ourselves. In the face of the obvious complexities of dealing with heterogeneous Web APIs, feeds in different formats and the like, we have found that distilling a simple, relational world model greatly facilitates model construction and provides a useful abstraction to simplify application logic. Based on our initial experiences, we believe that this approach to using diverse information sources on the Web to characterize the user's situation and activity will foster the creation of new, more personal applications and interfaces that can effectively adapt to individuals and their dynamically changing needs[8].

Acknowledgements

This work is funded in part by the National Science Foundation, Nokia Research, WSRI, and a Royal Academy of Engineering Senior Research Fellowship. We would like to thank our PLUM/PRUNE and List-it collaborators and student researchers, including Michael Bernstein, Jamey Hicks, Greg Vargas, Katrina Panovich, Paul André, and Brennan Moore.

5. REFERENCES

[1] Bernstein, M., Van Kleek, M., Karger, D., schraefel, mc, "Information Scraps: How and Why Information Eludes are Personal Information Tools" ACM Trans. Info. Systems, 26,4 (Sept 2008), 1-46.

[2] Budzik, J., and Hammond, K. Watson: Anticipating and Contextualizing Information Needs. In *Proc. American Society for Information Science and Technology 1999.*

[3] Dumais, S., Cutrell, E., Cadiz, J., Jancke, G., Sarin, R., and Robbins, D. Stuff I've Seen: A System for Personal Information Retrieval and Re-Use. In *Proc. SIGIR 2003*, ACM Press (2003).

[4] Jones, E., Bruce, H., Klasnja, P, Jones, W. I Give Up! Five Factors that Contribute to the Abandonment of Information Management Strategies. In *Proc. American Society for Information Science and Technology 2008*

[5] "Plazer" software from Plazes http://www.plazes.com.

[6] Rhodes, B.J. Margin Notes: Building a Contextually Aware Associative Memory. In *Proc. IUI 2000.*

[7] Time Expression Recognition and Normalization. http://timex2.mitre.org

[8] Van Kleek, M., Bernstein, M., Panovich, K., Vargas, G., Karger, D., schraefel, mc. Examining Personal Information Keeping in a Lightweight Note-Taking Tool. In *Proc. CHI 2009*, ACM Press (2009).

[9] Van Kleek, M., André, P., Karger, D., schraefel, mc. Mixing the reactive with the personal: Opportunities for end-user programming in Personal information Management. *To appear in EUP-WWW, End User Programming for the Web*, ACM Press, 2009.

[8] PRUNE and NTF are released under the MIT License and available for download at http://plum.csail.mit.edu.

Annotating URLs with query terms: What factors predict reliable annotations?

Suzan Verberne
CLST,
Radboud University Nijmegen
s.verberne@let.ru.nl

Max Hinne
Dept. Computer Science,
Radboud University Nijmegen
mhinne@sci.ru.nl

Maarten van der Heijden
Dept. Computer Science,
Radboud University Nijmegen
m.vanderheijden@cs.ru.nl

Eva D'hondt
CLST,
Radboud University Nijmegen
e.dhondt@let.ru.nl

Wessel Kraaij
Dept. Computer Science,
Radboud University Nijmegen
TNO, Delft
kraaijw@acm.org

Theo van der Weide
Dept. Computer Science,
Radboud University Nijmegen
tvdw@cs.ru.nl

ABSTRACT

A number of recent studies have investigated the relation between URLs and associated query terms from search engine log files. In [5], the query terms associated with the domain of a URL were used as features for a URL classification task. The idea is that query terms that lead to successful classification of a URL are reliable semantic descriptors of the URL content. We follow up on this work by investigating which properties of a URL and its associated query terms predict the classification success. We construct a number of URL and query properties as predictors and proceed to analyze these in-depth. We conclude that the classification success — and thus the reliability of the query terms as URL descriptors — cannot easily be predicted from properties of the URL and the queries.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information filtering

General Terms

Click data, URL classification, Human factors

1. INTRODUCTION

In previous work on the use of query log data [5], the authors investigated the applicability of semantic annotation of web pages by creating short document descriptions (term lists) extracted from associated queries. The assumption here is that, when presented to a user, these term lists may help in the disambiguation of a URL and/or identify whether the URL corresponds to the user's query intent [3]. The term lists in [5] were extracted from the (weighted) set of query terms that are associated with an URL. In order to find out whether the associated term lists provided good descriptions of the URL (and consequently, good clues in disambiguation), a classification experiment was conducted on a set of URLs, using the term lists as features. Depending on the level of query term aggregation, a classification accuracy of up to 45% was obtained [5].

These classification results are reasonably satisfying. We aim at a future implementation of semantic annotation of URLs in the user interface of a web search engine. In this implementation, the search engine not only has the query term descriptors for a URL available, but also the indexed content of the web page. Previous work on semantic annotations from URL content and query logs [11] showed that the query terms associated to a URL provide useful additional information to terms extracted from the content of the web page. Thus we can expect the classification accuracy to go up if we not only use the query terms but also the content terms.

However, a classification accuracy of 45% means that URL classification based on query terms only is unsuccessful for more than half of the URLs. In order to prevent the semantic annotation of URLs to be negatively influenced by unreliable query term descriptors, we aim to predict the reliability of the query term annotations given a URL and its associated queries. Therefore, in the current paper, we study which properties of the URL and the associated term list can predict the reliability of the query term annotations. Following [5], we consider the classification accuracy as indicator for the informativeness and reliability of the query terms annotation: the better a set of query terms describes a URL, the higher the chance that the URL is classified correctly. Thus, we investigate the relation between URL and query properties on the one hand and the classification quality on the other hand.

2. RELATED WORK

Other studies have shown that query–URL associations and click information can serve as a means for implicit feedback [4, 6, 7] and for learning to rank e.g.[1]. Several others have investigated whether a collection of queries and click information can be used as a model of the semantic contents of a web page [2, 8]. A document representation based on queries has been compared with a traditional document content vector space representation for a clustering task [9]. The query based representation resulted in a better clustering than the document content based representation. A similar experiment has been conducted where human assessors were asked whether they preferred a document description

SIGIR '09, July 19-23, 2009 Boston, United States.

based on queries or on a vector space representation [11]. The assessors tended to prefer the query-based representation.

Most related work uses site access logs of a portal page, which means that the log files show a complete picture of all queries leading to that site: query terms can be derived from the referring URL in the HTTP-header. In our case, we use a much larger collection of web pages and click data that is based on the query log of one single search engine (See Section 3.1). Our data do not comprise the page content of the URLs because the page content was not available in the query log data and recrawling would give many inconsistencies due to web pages changing significantly in the course of a few years. Our work differs from [5] since we explicitly aim to explain *in which cases* (for which types of URLs) query log data can be used to inform the user about the semantic contents of a web document, or for disambiguation of a URL or identification of query intent.

3. EXPERIMENTS

The objective of our experiments is to identify a set of key factors that predict whether a URL can be classified correctly using query log data. With the aid of such factors, a search engine can use terms from query log data as document descriptors, which help the user in disambiguating URLs or finding URLs that match the user's query intent.

3.1 Data

RFP: The Microsoft 2006 RFP[1] dataset consists of approximately 14 million queries from US users entered into the Microsoft Live search engine in the spring of 2006. For each query the following details are available: a query ID, the query itself, the user session ID, a time-stamp, the URL of the clicked document, the rank of that URL in the result list and the number of results.

DMOZ: The DMOZ Open Directory RDF Dump[2] is a set of URLs and their class labels according to the Open Directory Project DMOZ. E.g. `bikeriderstours.com` — `Top/ Sports/ Cycling/ Travel/ Tour_Operators`. We restricted the data to DMOZ level 2 labels (e.g `Top/ Sports`). We discarded the URLs labelled `Top/Regional`, since `Regional` is the top node of a different hierarchy (a regional classification). The intersection of the RFP and DMOZ collections (with the above restriction) consists of 245.742 URLs, distributed over 15 classes.

3.2 Properties of URL and query

In [5] the classification features for URLs in the RFP-DMOZ intersection were extracted by finding the query terms that were most strongly associated with the URL[3]. These features were aggregated at the level of the URL, the domain of the URL and the individual words in the URL. Using these features, the URLs were classified with Adaboost.MH [10]. The highest classification accuracy was achieved when the query terms were aggregated at the level of the domain of the URL. In the current paper, we therefore focus on query terms associated with URLs on the domain level, aggregating queries over all URLs from our data collection that belong to the same domain[4].

In order to investigate what properties of the URL and the associated query terms play a role in the correct classification of some URLs and the incorrect classification of others, we extracted the following properties for each URL in the RFP-DMOZ intersection:

D: The domain of the URL.

DL: The number of terms the domain was compounded of (the domain length)[5]

NC: The number of clicks in the RFP dataset that were associated with the domain.

NUQ: The number of unique query terms associated with the domain.

AQC: The average number of query terms per click on a URL from the domain.

MCP: The position that the clicked URL had in the result list of the search engine, averaged over all clicks that led to the domain.

PN: The proportion of navigational queries in the total number of queries that led to the domain of this URL. We consider a query to be navigational if the concatenated query terms are a substring of the URL string (e.g. "bike riders tours" is a navigational query for the URL `www.bikeriderstours.com`).

TWE: The token-wise entropy of the domain, as the sum of all the terms the domain is compounded of, i.e.: $H(D) = -\sum_{t \in D} P(t) \cdot \log P(t)$, with $P(t)$ the probability of observing term t in any domain in the RFP-DMOZ intersection.

KLD: The Kullback-Leibler divergence of the associated query term probability distribution, relative to the distribution of all query terms in the RFP dataset, i.e.: $D_{KL}(P\|Q) = \sum_{t \in D} P(t) \cdot \log \frac{P(t)}{Q(t)}$ with $P(t)$ the probability of observing t in all queries associated with D and $Q(t)$ the probability of observing t in all queries in the RFP collection.

For all but the first of these properties, we investigated their relation to classification success. Our hypothesis was that especially NC and NUQ would have a positive predictive value for the classification success. We expect that more clicks (higher NC) and more unique query terms (higher NUQ) for a domain result in a better representation of the domain and therefore in a better classification accuracy. The details of our analyses are in Section 4 below.

[1] `http://research.microsoft.com/en-us/um/people/nickcr/wscd09/`

[2] `http://rdf.dmoz.org/`

[3] Strength of association was calculated using Kullback-Leibler divergence with the total query collection as background model.

[4] As a consequence, we can only investigate URL properties that generalize to the domain level. Moreover, aggregating on the domain level has the risk of grouping together heterogenous URLs from large domains. We come back to this in Sections 4 and 6.

[5] We decompounded the domains using a script that subsequently looks up substrings in the CELEX lexicon (`http://www.ldc.upenn.edu/`) and greedily splits the domain string into lemmatized lexicon entries. E.g. the domain `bikeriderstours.com` was decompounded into the lemmas *bike*, *rider* and *tour*).

4. RESULTS

We considered three different strategies for finding the relevance of each of the predictors for the success of the classification: calculating the correlation coefficient ρ in order to get an indication of the strength and the direction of the relation between each predictor's value and the classification outcome. However, this coefficient assumes a linear relation that is independent of other predictors. Our data seemed more complicated than that. Therefore, we assessed the possibility of using a logistic regression model (LRM) for predicting the classification outcome based on the predictor values (normalized to their z-score). Unfortunately, the LRM outcome was difficult to interpret: We did get positive and negative predictor coefficients that significantly contributed to the prediction model but the model fit on the data was relatively poor.

These preliminary results suggested that there is no linear relationship between any of the predictors that we investigated and the classification success. We felt however that some tendency could be discerned from the individual predictors' values and the classification accuracies for specific ranges of these values. In order to assess this hypothesis, we created 10 bins for the values range of each predictor. Subsequently, we derived the classification accuracy for each bin, together with the number of domains in this range. We plotted these numbers in bar charts in order to visualize the relation between the value ranges of the predictors and the classification accuracy.

Unfortunately, we did not find very strong tendencies for most of the predictors that would support the idea that the classification success can be predicted from these properties. Most of the bar charts appeared to be relatively flat, confirming that the classification accuracy is relatively stable, only slightly dependent of the value of the predictor. As an example, Figure 1 shows the classification accuracy as a function of the token-wise entropy of the domain. The only bar that is rising above the others is the right-most one, representing the domains with the maximum entropy value. However, this bar only represents a small number of domains (3,423) and the classification accuracy for this range is still mediocre (60%).

In the next sub-section, we discuss the results for the two predictors that we had expected to give the most promising results (see Section 3.2).

4.1 Analysis of the NC and NUQ predictors

NC: When we look at the number of domains in relation to (a range of) the number of clicks on those domains (Figure 2), we first notice that most domains in our data collection have a small number of associated clicks (1 to 4). At the same time we see that domains with the lowest numbers of clicks are the domains with the lowest classification accuracy. This confirms our earlier assumption that many clicks result in a better representation of the domain and therefore a better classification accuracy. The maximum classification accuracy is 58% (for the range of 33–64 clicks). However, Figure 2 also shows that for a higher number of clicks, the classification accuracy starts to decrease again.

We suspect that this behavior can be explained from the heterogeneity of the domains that have a large number of associated clicks. For example, portal web sites such as `ebay.com` or `amazon.com` contain many URLs that may be very diverse in their semantic content. Consequently, these

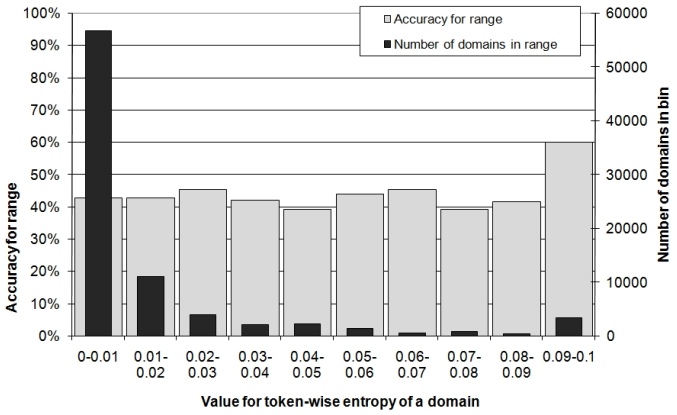


Figure 1: Classification accuracy as a function of the token-wise entropy of the domain. The token-wise entropy values have been grouped in ranges of i to $i + 0.01$ for $i \in \{0, \ldots, 0.09\}$

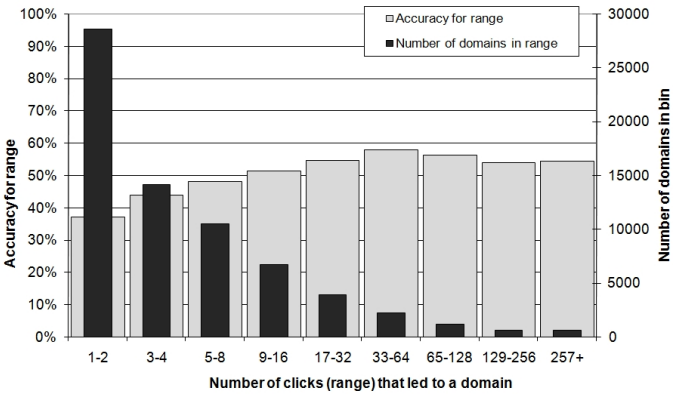


Figure 2: Classification accuracy as a function of the number of clicks that led to a domain. The numbers of clicks have been grouped in ranges of 2^i to 2^{i+1} terms for $i \in \{0, \ldots, 9\}$

URLs are harder to classify, since in the aggregated term set for the corresponding domain there are many terms for semantically unrelated URLs from the same domain.

NUQ: When we look at Figure 3, we see that the number of domains in a given range of unique query terms decreases much less sharply than for the number of clicks. However, we see a similar pattern in the classification accuracies for these ranges. For domains with 17–32 unique associated terms, the accuracy is optimal. Figure 3 shows that classification accuracy sharply increases initially for an increasing number of unique query terms, starting at 30% for domains with only 1–2 unique terms, up to 60% for domains with 17–32 unique terms. After that point, the accuracy decreases again.

Domains with very few unique terms apparently provide a too sparse classification vector to be classified correctly. At the other end of the spectrum, domains with too many unique terms are hard to classify as well. We again attribute this to the heterogeneity of the domains with a large number of unique query terms: it is very difficult to classify them as belonging to a single class.

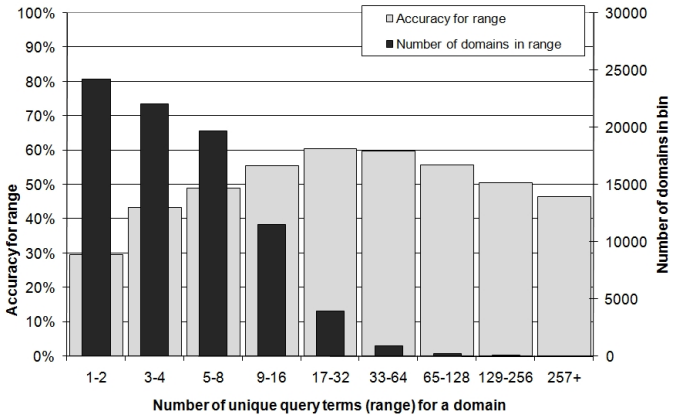


Figure 3: Classification accuracy as a function of the number of unique query terms per domain. The numbers of unique query terms have been grouped in ranges of 2^i to 2^{i+1} terms for $i \in \{0, \ldots, 9\}$

5. DISCUSSION

After analyzing our predictors in detail, we found that many of them cannot predict the classification success. The two most promising predictors (number of clicks and number of unique query terms) showed interesting tendencies but do not provide ranges of high accuracies (optimal ranges give 60% classification accuracy). It is clear that the success of classifying URLs based on query terms depends on many different factors. In the previous section, we mentioned the heterogeneity of the domain as a potentially important factor.

If we want to adapt our strategy for the heterogeneity of domains (for example, by not providing query-based descriptions for very heterogenous domains), the question that rises here is how we can identify domains as being heterogenous. Two of the factors that we saw in Section 4 are the number of clicks and the number of unique query terms that are associated with a domain. A third factor may be the domain size: the more URLs a domain contains, the larger the heterogeneity of the domain probably is. Part of our future work will be to estimate the domain heterogeneity based on these factors.

6. CONCLUSION AND FURTHER WORK

We continued the work of [5] and investigated which factors are relevant for the success of URL classification based on associated query terms. We created a series of classification success predictors and subsequently analyzed their relation to the classification success. None of the predictors we investigated can fully predict the classification success. We found however that a couple of predictors show interesting tendencies: the number of clicks on URLs (**NC**) and the number of unique terms associated with a URL (**NUQ**). In both cases, the predictors initially correlate positively with the classification accuracy, but after a certain saturation point this correlation becomes negative. We suggest that this is caused by heterogeneous domains (domains that contain URLs from different semantic categories). We argue that our suggested approach of providing query terms as document descriptors for disambiguation is particularly useful for URLs from homogenous domains.

An important point for further research is to determine the heterogeneity of a domain using query log data. Another direction is to investigate what factors predict classification accuracy when query terms are not aggregated on domain level, but on the level of individual URLs. As these cannot be heterogenous, it will be worthwhile to see the performance of the predictors in this situation.

We are currently experimenting with different types of classifiers in order to see whether we can improve the classification accuracy of our data. We also study our data in more detail in order to see whether removing a subset of the click data from the training set can increase the classification performance. This subset can be either category-based (remove noisy categories), feature-based (remove instances with too few query terms) or based on overall consistency (remove instances that have very similar term sets but contradictory classes).

In the somewhat more distant future, we aim to investigate the possibilities of implementing our URL descriptor approach in a user interface. Following the results obtained by [11], we will combine salient terms from the URL's content and the queries associated with the URL into a semantic annotation of the URLs in the result list. One challenge that we foresee for this experiment is the evaluation: User judgments are time-consuming but essential for this kind of implementation.

7. REFERENCES

[1] E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 19–26, New York, NY, USA, 2006. ACM.

[2] I. Antonellis, H. Garcia-Molina, and J. Karim. Tagging with queries: How and why? In *ACM WSDM '09*, 2009.

[3] D. J. Brenes, D. G. Avello, and K. P. Gonzalez. Survey and evaluation of query intent detection methods. In *Proceedings of WSCD '09*, pages 1–7. ACM New York, NY, USA, 2009.

[4] S. Fox, K. Karnawat, M. Mydland, S. Dumais, and T. White. Evaluating implicit measures to improve web search. *ACM Trans. Inf. Syst.*, 23(2):147–168, 2005.

[5] M. Hinne, W. Kraaij, S. Raaijmakers, S. Verberne, T. van der Weide, and M. van der Heijden. Annotation of URLs: more than the sum of parts. In *SIGIR '09: Proceedings of the 32th ACM SIGIR international conference on Information Retrieval*, New York, NY, USA, 2009. ACM.

[6] T. Joachims, L. Granka, B. Pan, H. Hembrooke, and G. Gay. Accurately interpreting clickthrough data as implicit feedback. In *Proceedings of the 28th annual international ACM SIGIR conference*, pages 154–161. ACM New York, NY, USA, 2005.

[7] D. Kelly and J. Teevan. Implicit feedback for inferring user preference: a bibliography. *SIGIR Forum*, 37(2):18–28, 2003.

[8] B. Krause, R. Jäschke, A. Hotho, and G. Stumme. Logsonomy - social information retrieval with logdata. In *Hypertext*, pages 157–166, 2008.

[9] B. Poblete and R. Baeza-Yates. Query-sets: using implicit feedback and query patterns to organize web documents. In *Proceedings of WWW '08*, pages 41–50. ACM New York, NY, USA, 2008.

[10] R. E. Schapire and Y. Singer. Boostexter: A boosting-based system for text categorization. *Machine Learning*, 39:135–168, 2000.

[11] M. van der Heijden, M. Hinne, W. Kraaij, S. Verberne, and T. van der Weide. Using query logs and click data to create improved document descriptions. In *Proceedings of WSCD '09*, pages 64–67. ACM New York, NY, USA, 2009.

Evaluating the Impact of Snippet Highlighting in Search

Tereza Iofciu
L3S Research Center
iofciu@L3S.de

Nick Craswell and Milad Shokouhi
Microsoft Bing Search
{nickcr,milads}@microsoft.com

ABSTRACT

When viewing a list of search results, users see a snippet of text from each document. For an ambiguous query, the list may contain some documents that match the user's interpretation of the query and some that correspond to a completely different interpretation. We hypothesize that selectively highlighting important words in snippets may help users scan the list for relevant documents. This paper presents a lab experiment where we show the user a top-10 list, instructing them to look for a particular interpretation of an ambiguous query, and track the speed and accuracy of their clicks. We find that under certain conditions, the additional highlighting improves the time to click without decreasing the user's ability to identify relevant documents.

1. INTRODUCTION

When users view a list of search results they see 'snippets' of text from the retrieved documents. A snippet helps the user decide whether to click, view and potentially make use of a document. A good snippet gives an indication of whether a document seems relevant, deserving click.

This paper evaluates *lists of* snippets, in the context of ambiguous queries. For ambiguous queries, a user may be faced with some results that are completely off-topic. For example, when users type the query 'house', they may be looking for information on the US House of Representatives, the TV series House or real estate. When users type 'microsoft' they may be looking for investment information, products to buy or technical support. There are multiple interpretations of the query, and it is unlikely that a user wants all of them. Therefore snippets should allow users to quickly reject results that are completely off topic, and scan towards those that are valuable. Therefore our experiments involve scanning a results lists of ambiguous queries.

In particular we consider two types of highlighting for the words in snippets. Our baseline approach is similar to the typical interfaces of the current web search engines, where the user's query keywords are highlighted in bold. Our other method highlights additional words (in yellow), that are not query words but are important for that particular document. The baseline method always highlights the same words in each snippet, while the new approach highlights the *differences* between snippets.

For example, for the query "Cornwall England", where the query intent is not very clear, a search engine retrieves general information pages, like Wikipedia pages, but also pages with tourist information. The baseline highlighting puts only the words 'cornwall' and 'england' in bold. Our new method, in addition, highlights 'tourist', 'Wikipedia' and 'pictures'. This potentially allows, for example, a user who is ready to book their holiday to find travel booking sites more easily. In one experiment the additional highlighting is automatic, in the other it is manual. In both cases the hypothesis is that users will be able to scan towards relevant documents more quickly with the additional highlighting.

2. RELATED WORK

There are many studies in literature focusing on different aspects of document representation and summarization in the context of information retrieval. Some approaches are evaluated in a task-oriented manner where speed and accuracy are compared for different search result representations. A recent example of 'extrinsic' evaluation, with references to past studies, is [1].

Alternatively snippet evaluation can be intrinsic: For example measuring whether the summary contains important n-grams from the document. These measures, such as DUC's ROUGE[1], are correlated with extrinsic measures, and have the advantage of being reusable. The present study is non-standard, so we can not repeat any existing intrinsic or extrinsic method. Ours is an extrinsic evaluation concerned with *lists* of summaries.

Our study is similar to the one presented in [5] and later in [4], where the importance of query biased summaries for web search result representation was demonstrated. A task-oriented evaluation was conducted, similar to [1], where the participants had to fulfill different types of search tasks. In the task-oriented studies the users were free to build their own queries in order to solve the tasks. Similar to our experimental setup, in [3] the queries, TREC topics in this

[1] `http://berouge.com/`

case, and their search results, have been fixed thoughout the experiment.

3. USER STUDY SETUP

This paper describes two rounds of experiments. The main difference between the two is the highlighting method (manual vs automatic) and the method for selecting ambiguous queries. However, we made a number of general improvements in our second experiment.

In both experiments our experimental subjects followed a similar procedure. The user is shown an ambiguous query, along with a 'topic description' of how the query should be interpreted. For example, the query 'house' and the description 'information on the TV show'. Then, the user clicks a link to indicate that they are ready, and we show the top-10 list for the query (taken from the Microsoft Web search engine). The user's task is to identify and click a document that fits the topic description, and then the move on to the next query-topic description. The top-10 results and snippets are always the same for each query, and query words are always highlighted in bold. We only vary whether there is additional highlighting, in yellow, of non-query words.

3.1 Manual Experiment Setup

Our pilot experiment used manual highlighting rather than any realistic method for automatically highlighting extra words in snippets. We describe the manual experiment, although the 'automatic highlighting' experiment improves on it in a number of dimensions.

Selecting the queries. If a query has most of its clicks on a single URL, it is probably not an ambiguous query. It is more likely to be navigational [2]. To select ambiguous queries we first select queries with skewness smaller than 0.5, from the 'torso' of the query distribution (not a head query, not a tail query). We manually inspected the top-10 list for 100 of these queries, to identify 50 that seemed to have results that cover more than one topic, and used these as our manual experiment queryset.

Query intent. For each of the selected 50 queries, we developed a topic description. The topic was selected to describe some aspect of the query's top-10 results. We also judged the relevance of each result to the topic, and made a second pass where topics and judgments were checked by a second assessor.

Highlighting. Three assessors each viewed the top-10 result snippets and selected 'important' words for highlighting. The result snippets were shown in the order they were retrieved by the search engine. They did so without knowing the query's topic description, to avoid any bias towards that interpretation. In our experiment, we then highlighted any word or phrase that was selected by two or more assessors.

3.2 Automatic Experiment Setup

After the manual experiment, we noticed that some queries were not really ambiguous (for example 'comet 17p holmes'). This is a problem because it led to the development of a contrived topic, which was confusing to our users and unlikely to agree with our highlighting. In our second experiment, we improved our method for selecting ambiguous queries and introduced an automatic highlighting method.

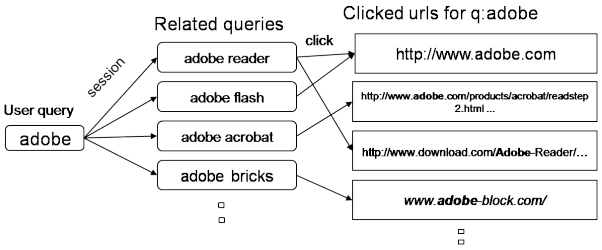


Figure 1: Ambiguous query and intent selection.

Selecting the queries and query intents. To help us identify ambiguous queries, we developed a distinctiveness measure for search results based on information from search logs. Session information connects query q and query q' if query q tends to be followed by q' within user sessions. Click information connects query q and URL u if we have observed users clicking on search result u for query q.

To calculate our distinctiveness score for a query, such as 'adobe' in Figure 1, we assign queries to the top-10 URLs. The assignment is according to click data, however we only include queries that are also connected to the original query in session data. The query 'adobe bricks' has a click connection with one URL, and a session connection with 'adobe', so it is associated with the URL.

The distinctiveness of a URL is the proportion of its associated queries that were not assigned to any other URL. The output of our process is a set of query-URL pairs with distinctiveness of 0.5 or greater.

For the automatic experiment, 40 pairs of query and distinct URL were manually selected from 700 candidates. The query's 'topic description' was 5 of the associated click/session queries, preferring queries with greater numbers of clicks.

Highlighting. We used three approaches for automatic highlighting:

- Top query phrase. Using click data only (not session data) we highlighted the most popular click query that occurred in the snippet, if any.
- Top URL anchor phrase. If no query phrase was highlighted, we highlighted the most popular incoming anchor phrase that occurred in the snippet. Anchor information came from a large Web search engine.
- Wikipedia disambiguation terms. Where a Wikipedia disambiguation page existed for a given query, such as "Cornwall (disambiguation)"[2], then all the disambiguating entity names were highlighted in the query result page.

[2]`http://en.wikipedia.org/wiki/Cornwall_(disambiguation)`

Figure 2: Automatic highlighting for the query "Cornwall".

The first two approaches can highlight differently for each result in the top-10, since each URL has different click data and incoming anchor text. The third approach was applied globally to the search results.

Figure 2 shows an example of automatic highlighting. As always, the additional highlighting gives the highlighted word/phrase a yellow background.

4. EXPERIMENT RESULTS

In both experiments, each user saw all queries. Half the users saw additional highlighting on the odd numbered queries. The other users saw it on even numbered queries. At the end of the experiments the participants were asked to answer a questionnaire.

4.1 Manual Experiment

The manual experiment had 16 participants who each processed 50 queries. We manually judged the relevance of each top-10 result with respect to the chosen interpretation (topic). The same top-10 was also used for topic development (i.e. assigning the desired topic to a query), so upon judging the top-10 there were always one or more relevant documents found for the assigned topic. Figure 3 shows that relevant documents were distributed evenly over ranks, but users tended to click documents near the top of the list. This is consistent with our instructions to click the first relevant document found. It also matches the ranks of the 'shallowest relevant document' for each query, i.e. the first relevant document to be found in the top-10 retrieved.

Results indicate that manual highlighting was not useful. Table 1 shows that users were slower when faced with the new highlighting, and users delayed longer in cases where they eventually clicked an irrelevant document. We then divided our observations into two groups, fast and slow, based on the time to click. We show the accuracy of clicks in Table 2. This again indicates that a delay in the manual highlighting case is associated with making more mistakes.

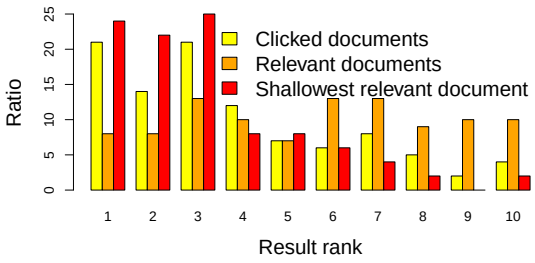


Figure 3: Relevant results vs. clicked results

Table 1: Average time until click

Highlighting	Time (sec) when relevant	Time (sec) when not relevant
baseline	20.83	19.19
manual	23.24	27.38

Table 2: Probability of clicking a relevant result

Highlighting	Relevance (when fast)	Relevance (when slow)
baseline	0.76	0.79
manual	0.78	**0.67**

4.2 Automatic Experiment

The automatic experiment had 8 users who each processed 40 queries. Having identified a number of problems in the manual experiment, we made a number of changes in the automatic experiment. Of course we employed an automatic highlighting method and used a new method for identifying potentially ambiguous queries (see Section 3.2). For each query users now click the topic description itself to indicate that they are ready to see the top-10. This was intended to reduce the chances of a user ignoring a topic. We also precomputed and optimized the HTML of top-10 lists, to make the top-10 lists render on the screen more quickly.

Highlighting had a much smaller effect in the automatic experiment than in the manual experiment. In particular, automatic highlighting did not cause users to become both slow and inaccurate for some queries. For example, adding automatic highlighting did not change the click distribution over ranks (Figure 4). The automatic method highlighted fewer words than the manual method, and may have been more consistent.

In the automatic experiment click accuracy was 0.9, compared to 0.75 for the manual experiment. In the automatic experiment, this level of accuracy was maintained with and without the additional highlighting. A breakdown of accuracy differences per-query is presented in Figure 5.

Within the automatic experiment, the main effect we observed was the time taken to click. The baseline highlighting had a time till click of 13.5 seconds, while the time for automatic highlighting was 11.2 seconds. Figure 6 shows the

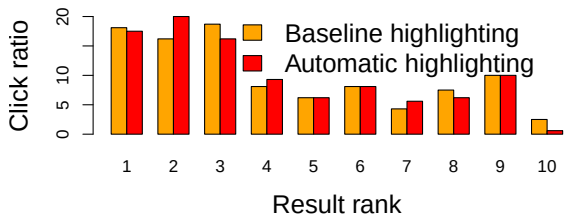


Figure 4: Click histogram highlighting vs. baseline highlighting

difference in average time on a per-query basis.

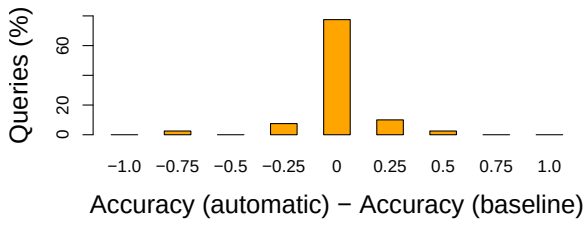


Figure 5: Accuracy of automatic vs. baseline highlighting

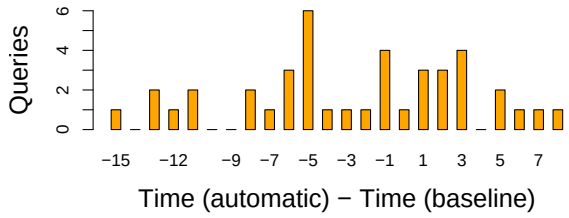


Figure 6: Time taken for automatic vs. baseline highlighting

4.3 Questionnaire Results

At the end of the experiment the participants had to fill in a questionnaire about the search tasks and their experience with the experiment. In the manual experiment users were more likely to say that there was too much yellow highlighting (the additional highlighting was always yellow).

In both setups more than 60% of the participants have reported to having been sometimes familiar with the search topics and more than 70% found the connection between the query and the selected intent often understandable.

5. CONCLUSION AND FUTURE WORK

This paper described our experiments in highlighting the important words in the search snippets for ambiguous queries. Unlike many summarization experiments, we tested how easy it was to scan a top-10 list of snippets, rather than the quality of individual snippets.

Our manual experiment was set up with a lot of human effort: Manual topic development, manual highlighting of the snippet words selected by two out of three assessors, and full relevance judgments of the top-10s. However, we suspect that some topic descriptions were somewhat 'contrived', having been developed for queries that were not really ambiguous. This may have been confusing our users, who also reported in the post-experiment questionnaire that there was too much highlighting. Overall, showing manual highlighting was associated with slower and less accurate clicks.

Our automatic experiment used a log analysis method to identify queries that seem ambiguous, because they have one distinctive URL in the top-10. Although this set of query-URL pairs still required manual vetting, we believe it was a much cleaner set of ambiguous queries. We also introduced an automatic highlighting method based on click logs, anchors and Wikipedia disambiguation pages. Finally we made two changes to the experimental interface, by speeding up the software and increasing the focus on topic descriptions by forcing them to click the description before proceeding. In combination, these changes led to us no longer seeing slow and inaccurate click behavior in the presence of highlighting. Instead, click accuracy was maintained, while speed improved by 17%, to about 11.2 seconds per query.

One drawback of our experiments is that we only used ambiguous queries, and there was always a manual vetting procedure during query selection. Therefore we have not studied the influence of highlighting in general. In future work we would like to understand the influence of query type on our experiments, and improve our automatic techniques for discovering ambiguous queries, since it may be desirable to highlight differently for different query types. We also intend to experiment with eye-tracking tools, to measure more directly the influence of highlighting on user attention.

6. REFERENCES

[1] Hideo Joho and Joemon M. Jose. Effectiveness of additional representations for the search result presentation on the web. *Inf. Process. Manage.*, 44(1):226–241, 2008.

[2] Uichin Lee, Zhenyu Liu, and Junghoo Cho. Automatic identification of user goals in web search. In *WWW '05*, New York, USA, 2005. ACM.

[3] Anastasios Tombros and Mark Sanderson. Advantages of query biased summaries in information retrieval. In *SIGIR '98*, New York,USA, 1998. ACM.

[4] Ryen White, Joemon M. Jose, and Ian Ruthven. A task-oriented study on the influencing effects of query-biased summarisation in web searching. *Inf. Process. Manage.*, 39(5):707–733, 2003.

[5] Ryen White, Ian Ruthven, and Joemon M. Jose. Web document summarisation: A task-oriented evaluation. In *DEXA '01*, Washington, DC, USA, 2001.

Using Domain Models for Context-Rich User Logging *

Stephen Dignum
School of Computer Science and Electronic Engineering
University of Essex
Colchester, United Kingdom
sandig@essex.ac.uk

Yunhyong Kim
School of Computing
Robert Gordon University
Aberdeen, United Kingdom
y.kim1@rgu.ac.uk

Udo Kruschwitz
School of Computer Science and Electronic Engineering
University of Essex
Colchester, United Kingdom
udo@essex.ac.uk

Dawei Song
School of Computing
Robert Gordon University
Aberdeen, United Kingdom
d.song@rgu.ac.uk

Maria Fasli
School of Computer Science and Electronic Engineering
University of Essex
Colchester, United Kingdom
mfasli@essex.ac.uk

Anne De Roeck
Department of Mathematics and Computing
Open University
Milton Keynes, United Kingdom
a.deroeck@open.ac.uk

ABSTRACT

This paper describes the prototype interactive search system being developed within the AutoAdapt project[1]. The AutoAdapt project seeks to enhance the user experience in searching for information and navigating within selected domain collections by providing structured representations of domain knowledge to be directly explored, logged, adapted and updated to reflect user needs. We propose that this structure is a valuable stepping-stone in context-rich logging of user activities within the information seeking environment. Here we describe the primary components that have been implemented and the user interactions that it will support.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Query Formulation; H.5.2 [**User Interfaces**]: Natural Language; I.2.7 [**Natural Language Processing**]: Text Processing

General Terms

Domain Model, Graph Traversal, User Logging

1. INTRODUCTION

Searches within document collections like intranets differ from those within the general World Wide Web [6]. The terminology, structure, and services provided within an intranet are selected to meet organisational requirements, and, consequently, a considerable amount of time is spent by users trying to learn the domain characteristics even before they are able to identify the adequate questions to be submitted to a search system. In the AutoAdapt project, we hope to analyse and accelerate this learning process by implementing a system that presents and logs several domain model representations in response to each stage of a user's logged search activity. By encouraging and logging the direct interaction of users with domain model representations, collective domain user behaviour can be understood within context. The analysed user needs can be incorporated back into the system to adapt domain knowledge representations that are presented to the users, creating a continuous feedback loop.

Provision of domain model knowledge has been shown to aid user search for the information they need [3]. A domain model is effectively a structure that characterises the domain dataset from the domain user perspective, e.g. a graph where nodes represent domain concept terms and edges between nodes their relationship, possibly weighted to express how specific the term is or how closely related the terms are within the collection.

One of the difficulties in using traditional logging of user activity, such as submitted query terms, URL clicks, and page viewing time, to adapt search systems is the lack of sufficient context for identifying the user actions that are truly relevant to the user's information need. We implement methods of explicitly visualising domain models to accompany each search step, in addition to a list of links to search results, and a set of query term suggestions. By concurrently logging user interaction with the these components we have a mechanism to enable subsequent weblog analysis. For example, different document selections following the exploration of the same path may indicate relevance between documents, different paths leading to the same document may indicate relationships between paths, a comparison of path before and after a document selection should yield some understanding of the nature of the document selection.

We present here a working system including a graphical do-

SIGIR'09, July 19-23, 2009, Boston, USA.
[1]http://AutoAdaptProject.org

main model presentation, a document list and term suggestions designed to capture the described information.

2. RELATED WORK

It is frequently pointed out that users are reluctant to leave any explicit feedback when they search a document collection. However, implicit feedback, e.g. the analysis of log records, has been shown to be good at approximating explicit feedback. For example, users often reformulate their query and such patterns can help in learning an improved ranking function [2]. The same methods have shown to improve an adaptive domain model on a local Web site created using formal concept analysis lattice structures [4].

It has already been evidenced that users want support in selecting search words for query formulation but also it has been recognised that they want to stay in control with respect to making the final decision to submit a query [8]. Furthermore, it has been noted that users like to be provided with system-guided query suggestions even if suggestions are not relevant to the current query [7]. Users have shown signs of being more inclined, in a search environment that supports navigation, to submit new queries, or resubmit modified queries, than to navigate away from the result set [5]. Finally, increased activity in developing interactive features in search systems used across existing popular Web search engines suggests that interactive systems are being recognised as a promising next step in assisting information search. The work proposed in this paper is very much in line with what Belkin calls the *challenge of all challenges* in IR at the moment, to move beyond the limited, inherently non-interactive models of IR to truly interactive systems [1].

3. USER INTERFACE

In figure 1 we can see a screenshot of our demonstration system. There are four basic components, a) a simple entry box for query terms, b) a list of URLs with associated snippets, c) a graph displaying a segment of a domain model, and d) a list of suggested terms for query refinement.

The user enters a set of query terms, this results in a number of documents being returned. Using the query terms, additional terms are automatically extracted, e.g., from the domain model and the highest ranked documents. These terms are then represented as nodes in a graph as a segment of the domain model. The user can then traverse the graph by clicking on the nodes (the effect is to make that term the centre of the graph). On term selection the list of suggested terms is updated to show terms closely related to the selected term. The user can then add the term to the existing query or use it as a new query.

The modular nature of the software allows a standard user interface and logging structure (described in the next section) irrespective of the domain model creation and adaptation algorithms employed. We can, for example, examine different interaction styles and evaluate other domain model visualisation tools.

4. LOGGING INFORMATION

In addition to logging user query terms with presented and selected URLs it was decided to log the segment of the domain model presented to the user. As we intend to modify the domain model over time based on responses to the model presented, it is essential that a complete copy of the presented model segment is retained in the database. Of particular interest is the term positioned at the centre of the graph and the co-ordinates of the other terms. Using this information and the term clicks we can determine how the model was traversed, allowing us to identify which terms were also visible and ignored. Suggested terms (derived by the model) are also recorded along with any selection (to expand, or replace initially submitted query terms).

The logging structure allows us to record a number of user decisions without the need for explicit feedback. For example, the selection of a term in a domain model can provide a ranking of terms, i.e., above those shown but not selected. Also, suggested terms derived from a particular traversal can be ordered. In addition, we can compare sessions that have resulted in the same URL being selected in order to capture related terms or similar portions of the domain model. It is also possible to compare portions of different domain models to discover missed relationships or terms.

5. FUTURE WORK

As the next step, we propose to test the infrastructure in this document across several domain collections and model creation/adaptation algorithms to extensively evaluate the effectiveness of the system in capturing the context of user interaction.

6. ACKNOWLEDGEMENTS

AutoAdapt is funded by EPSRC grants EP/F035357/1 and EP/F035705/1. The JIT visualisation toolkit[2]was used for the domain model visualisation.

7. REFERENCES

[1] N. J. Belkin. Some(what) grand challenges for information retrieval. *SIGIR Forum*, 42(1):47–54, 2008.

[2] T. Joachims and F. Radlinski. Search engines that learn from implicit feedback. *IEEE Computer*, 40(8):34–40, 2007.

[3] U. Kruschwitz and H. Al-Bakour. Users Want More Sophisticated Search Assistants - Results of a Task-Based Evaluation. *JASIST*, 56(13):1377–1393, November 2005.

[4] D. Lungley and U. Kruschwitz. Automatically maintained domain knowledge: Initial findings. In *Proceedings of ECIR*, pages 739–743, 2009.

[5] M. Mat-Hassan and M. Levene. Associating Search and Navigation Behavior Through Log Analysis. *JASIST*, 56(9):913–934, 2005.

[6] M. White. *Making Search Work: Implementing Web, Intranet and Enterprise Search*. Facet Publishing, 2007.

[7] R. W. White, M. Bilenko, and S. Cucerzan. Studying the Use of Popular Destinations to Enhance Web Search Interaction. In *Proceedings of SIGIR'07*, pages 159–166, Amsterdam, 2007.

[8] R. W. White and I. Ruthven. A Study of Interface Support Mechanisms for Interactive Information Retrieval. *JASIST*, 57(7):933–948, 2006.

[2] `http://blog.thejit.org/javascript-information-visualization-toolkit-jit`

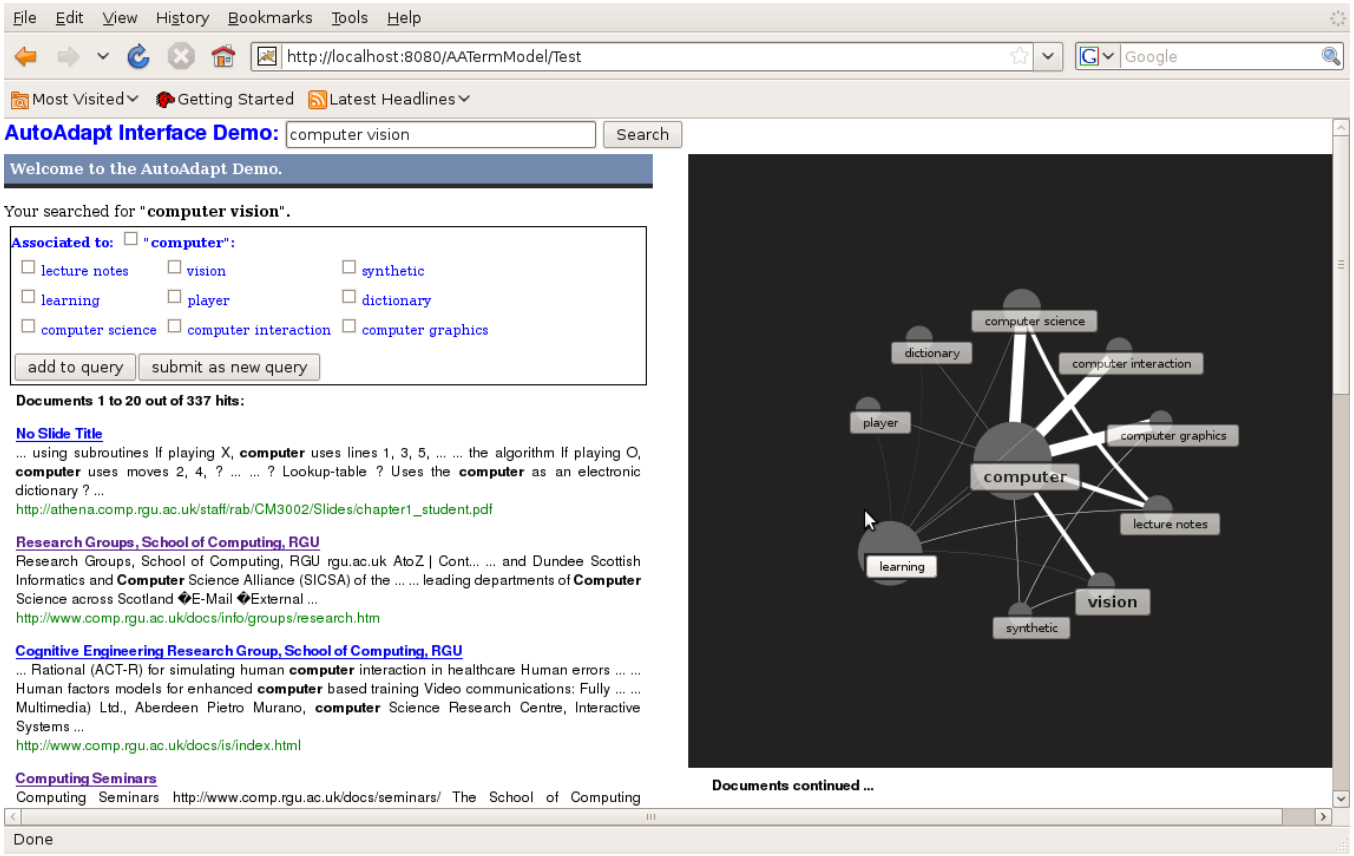


Figure 1: Screenshot of AutoAdapt Demo System.

Catching the User - User Context through Live Logging in DAFFODIL

Claus-Peter Klas
FernUniversität in Hagen
claus-peter.klas@fernuni-hagen.de

Matthias Hemmje
FernUniversität in Hagen
matthias.hemmje@fernuni-hagen.de

ABSTRACT

This demo will present the logging facilities to capture the user context within the Daffodil framework during a live search in computer science data sources. We propose to use the Daffodil system as an experimental framework for the evaluation and research of interactive IR. The system already provides a rich set of working services and available information sources. These services and sources can be used as a foundation for further research going beyond basic functionalities. In addition, the system can easily be extended regarding both services and sources. Daffodil's highly flexible and extensible agent-based architecture allows for easy integration of additional components and access to all existing services. Finally, the system provides a user-friendly graphical interface and facilitating services for log generation and analysis. The experimental framework can serve as a joint theoretical and practical platform for the evaluation of DLs, with the long-term goal of creating a community centered on interactive IR and DL evaluation.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: User Interface, Logging; H.3.7 [**Digital Libraries**]: User Issues

General Terms

information retrieval, visualization, interactive systems

1. INTRODUCTION

It is our intention to demonstrate the Daffodil system with regard to capturing the user context by logging all user initiated actions. Due to the rich functionality of the DAFFODIL system the user is able to explore the search domain in a comprehensive and sustainable way. During the interaction of the user with the system we capture the behavior through logging the actions. We categorized these actions into ten different conceptual events described in [1].

1.1 Daffodil-Framework

DAFFODIL consists of several tools to support the users search tasks. We would like to demonstrate the DAFFODIL [1] framework as an experimental system for the evaluation of the interaction in information search and retrieval. DAFFODIL is a virtual digital library system providing access to many sources from the domain of computer science, and targeted at strategic support of users during the information seeking and retrieval process. It provides basic and high-level search functions for exploring and managing information objects including annotations over a federation of heterogeneous digital libraries (DLs) based on a service-oriented architecture. For structuring the functionality, we employ the concept of high-level search activities for strategic support. A comprehensive evaluation in [2] showed that the system supported most of the information seeking and retrieval aspects needed for a computer scientist's daily routine.

Additionally DAFFODIL incorporates the concepts of adaptivity [3], collaboration, recommendation and awareness. In order to enable adaptivity and recommendations DAFFODIL collects implicit and explicit user interactions and system actions as described in previous publications [1]. This interaction can be examined and captured at various levels of abstraction, starting at the system/hardware level and covering the complete spectrum of user-system interaction.

2. LEVELS OF LOGGING

When using transaction logs for evaluation, the primary levels surveyed are the **user**, the **system**, and the **content** that is being searched, read, manipulated, or created. Because interaction between the system and the user can be captured at various levels of abstraction we focus on three levels of evaluation:

User behavior level: Data about users and their behavior are located at this level. Each user has a task to accomplish, within a certain social environment, and brings her individual knowledge to that task.
Concept level for comparative evaluations: The concept level captures data about generalized events generated by the DL user. By logging these events, user evaluation can be backed up with statistical data and a comparative evaluation of different users, systems and system content can be undertaken.
System level: System events happen on the computer or in the computer network where DL services are executed. This

[1] `http://www.daffodil.de`

level aggregates specific information concerning the state of the DL (e.g., database conditions, server load, or amount of network traffic) and its response (e.g., response time).

Through logging events by level, we have a horizontal view that tracks a sequence of events dealing with a single aspect of the DL. For example, by focusing on events that occur on the concept level, we can identify the user's moves and tactics as she works her way through the document space. In contrast, a vertical view across levels gives us information about the impact of a specific event across the DL system, from information about user behavior on the highest level to system specific data on the lowest level.

3. EVENTS ON THE CONCEPT LEVEL

At the concept level, we have identified several general event types that support comparative evaluation across DLs. Our focus on the concept level represents the centrality of these events for log analysis and interpretation: events that occur on the concept level indicate critical aspects of the user's interaction with the DL system and supply valuable data for rich interpretation of user behavior. As is highlighted in other DL logging studies [4], current approaches are often inadequate for capturing complex or abstract actions by the user and are therefore unable to elicit meaningful conclusions. By logging data about general event types at the concept level, we provide a basis for comparative evaluation across DLs and still gain insides into the users behavior.

The event types and event properties that we have identified are neither fixed nor complete and should be viewed as recommendations. that can also serve as discussion points in the community.

We have identified the following events on the concept level:
Search: The user formulates a *query* or *filter condition* that is to be processed by a given DL service against a *collection*.
Navigate: The user initiates an event by selecting from a set of possible moves from one point to another.
Inspect: The user accesses the details of a single object.
Display: The display event describes a specific *visualization* of the information presented to the user.
Browse: The user selects an event for viewing a set of DL objects (e.g., viewing a result list following a search).
Store: The user files an object for later reuse, either at a generic location (e.g., on a clipboard) or at a specific location (e.g., in a specific folder).
Annotate: The user adds information to an existing DL object, either by marking specific parts of it, by linking it to other digital library objects, or by adding inline or external comments.
Author: The user creates a new DL object or edits an existing object such as a document or annotation.
Help: The user requests help or information. The help event may be general or context-specific and can include introductory overviews or tutorials about the DL system.
Communicate: Users collaborate through communication, either by posing a simple question or through the use of specific tools or services.

4. DEMONSTRATION

In the demonstration we will present the DAFFODIL system with a specific focus on the live logging of user events.

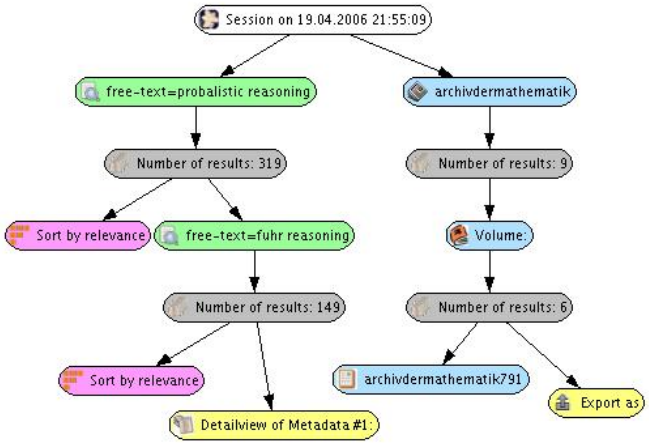


Figure 1: top-down tree visualization

Based on a given task several search and browse tools will be presented and a analysis, e.g. for relevance feedback will be given. In figure 1 a small search session if graphically presented as tree visualization. The colors of the nodes in the figure correspond to the concept level.events for easier recognition.

This captured information represents the basis to further understand and support the user. It of course does not excuse from running a real user evaluation. Such support could be done through recommendation via implicit relevance feedback as well as collaborative recommendations through other users in a similar situation. We think, that given the context model within the Daffodil-Framework, we are able to understand and categories user behavior and provide solid data to support system oriented IR evaluation, e.g. based on user simulation.

5. ACKNOWLEDGMENTS

This work is funded via the German Science Foundation (DFG) in the project LACOSTIR.

6. REFERENCES

[1] C.-P. Klas, H. Albrechtsen, N. Fuhr, P. Hansen, S. Kapidakis, L. Kovács, S. Kriewel, A. Micsik, C. Papatheodorou, G. Tsakonas, and E. Jacob. A logging scheme for comparative digital library evaluation. In *ECDL 2006*, Lecture Notes in Computer Science, pages 267–278, Heidelberg et al., September 2006. Springer.

[2] C.-P. Klas, N. Fuhr, and A. Schaefer. Evaluating strategic support for information access in the DAFFODIL system. In *ECDL 2004*, Lecture Notes in Computer Science, Heidelberg et al., 2004. Springer.

[3] C.-P. Klas, S. Kriewel, and M. Hemmje. An experimental system for adaptive services in information retrieval. In *Proceedings of the 2nd International Workshop on Adaptive Information Retrieval (AIR 2008)*, October 2008.

[4] B. Pan. Capturing users behavior in the national science digital library (nsdl). Technical report, NSDL, 2003. `http://dlist.sir.arizona.edu/848/`.

Massive Implicit Feedback: Organizing Search Logs into Topic Maps for Collaborative Surfing

Xuanhui Wang and ChengXiang Zhai
Department of Computer Science
University of Illinois at Urbana-Champaign
{xwang20, czhai}@illinois.edu

1. INTRODUCTION

Current search engines heavily emphasize on direct querying which tends to work well only for simple information needs such as navigational queries. However, direct querying may not support complex information needs such as exploratory search well [4, 11] since users' interactions are mainly limited to submitting a query, viewing results, and reformulating queries [1]. As a complementary way of information seeking with querying, browsing can be very useful for exploratory search or information foraging [5]. Unfortunately, with the current search engines, browsing is mostly limited to following hyperlinks or navigating through structures consisting of a fixed set of categories or other metadata available [4, 2].

We have been developing a new collaborative surfing system to enable users to go beyond hyperlinks to browse flexibly for *ad hoc* information needs. Our main idea is to view search logs as information footprints left by users in navigating in the information space and organize these footprints into a multi-resolution topic map. The map makes it possible for users to navigate flexibly in the information space by following the footprints left by other users. As new users use the map for navigation, they leave more footprints, which can then be used to enrich and refine the map dynamically and continuously for the benefit of future users. Thus, by turning search logs into a topic map, we can establish a sustainable infrastructure to facilitate users to surf the information space in a collaborative manner. Preliminary experiment results show that the topic map is effective in helping users to satisfy exploratory information needs [8]. In the following, we describe our system in more detail and discuss its potential impact on understanding users for improving information seeking.

2. SYSTEM DESCRIPTION

Figure 1 shows the interface of our system which is implemented as a meta-search engine interacting with Google. The interface has three panes: (1) The top pane is a querying box, where a user can submit a keyword query. (2) The left pane shows a portion of a multi-resolution topic map built based on search logs, where a user can click on a node to navigate into a topic region. (3) The right pane displays information corresponding to a topic region, including the clickthroughs made by previous users when they visit the topic region and the documents covered by the topic region.

These three panes allow a user to navigate in the information space in large, medium, and small steps, respectively. With the query box on the top, a user can make a long distance navigation into any topic region (i.e., "large steps"); with the topic map on the left pane, the user can navigate into related topic regions (i.e., "medium steps"); and with the display of a topic region in the right pane, the user can navigate by following hyperlinks (i.e., "small steps"). A user can take any of these three navigation actions at any time. Thus our system implements a unified information seeking model where both querying and browsing are viewed as ways to navigate in the information space.

Inside the system, when a user submits a query, the system would display the most relevant part of the topic map on the left pane and show the search results from Google for the query. When the user navigates on the map to click on a node (corresponding to a topic region), the system would automatically update the right pane to show corresponding search results using a query constructed based on the node selected by the user. In general, the right pane is always synchronized with the left pane to show the documents corresponding to the current node on the map.

The topic map promotes browsing and can naturally support exploratory search. For example, a user who wants to arrange a house can start with a query "table," zoom into "dinning table," zoom out to "dinning,", move horizontally to "kitchen," and further move to "appliance." From "table", this user can also horizontally move to "chair," to "desk,", or to "tablecloth." Another example is "wedding." From "wedding," we can zoom into different aspects of wedding such as "wedding dress,' "wedding vows," etc. We can also horizontally move to "vacation," "honeymoon," or "hotels." All these browsing traces can be leveraged to infer users' underlying information needs and better serve users with complex exploratory information needs. The browsing logs can be leveraged to improve the map and further help future users who have similar information needs.

A main technical challenge in developing this system is to construct topic maps. Currently, the nodes in topic maps are valid queries in search logs. All queries with the same number of keywords belong to the same level. The children

SIGIR'09, July 19-23, 2009, Boston, USA.

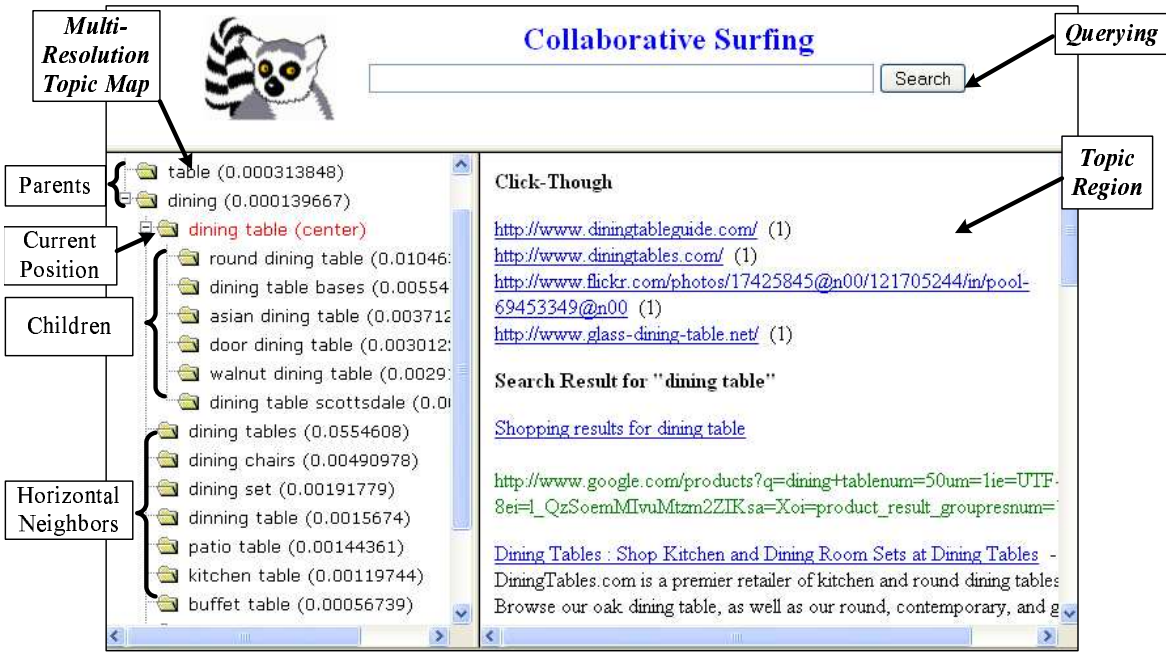


Figure 1: Interface snapshot of the demo system.

of a map node is obtained by adding a keyword into the current query and the neighbors of the query is by substituting a keyword in the current query. All these surrounding nodes are ranked accordingly. Specifically, we rely on the term co-occurrence in search logs to construct such a map and all the technique details can be found in [10].

3. MASSIVE IMPLICIT FEEDBACK

From the viewpoint of understanding users and exploiting user information to provide better search support, our system implements a strategy of massive implicit feedback [3, 7, 6, 9], where query logs and browsing logs of all users would be captured and leveraged to provide better support for future users in both querying and browsing. Indeed, the implicit feedback information collectable by the system includes not only the queries and clickthroughs available in a current search engine but also the browsing traces left by users in using the map. The system treats all these different kinds of user information uniformly as "information footprints" left by users and organizes them into a topic map to deliver benefits to future users. At the same time, new users would leave new footprints to allow the system to grow continuously over time to improve its support for browsing and querying. Thus, the system enables collaborative surfing where users help each other through sustained massive implicit feedback.

We hope our demo can stimulate discussions about many interesting questions related to the workshop: (1) How should we evaluate topic maps? (2) How should we evaluate such an interactive system? (3) How can we formally model a user based on both query logs and browsing logs? (4) How can we leverage maps to clarify user interests?

4. REFERENCES

[1] N. J. Belkin, S. T. Dumais, J. Scholtz, and R. Wilkinson. Evaluating interactive information retrieval systems: opportunities and challenges. In *CHI Extended Abstracts*, pages 1594–1595, 2004.

[2] M. A. Hearst. Clustering versus faceted categories for information exploration. *Commun. ACM*, 49(4):59–61, 2006.

[3] D. Kelly and N. J. Belkin. Display time as implicit feedback: understanding task effects. In *Proceedings of ACM SIGIR 2004*, pages 377–384, 2004.

[4] G. Marchionini. Exploratory search: from finding to understanding. *Commun. ACM*, 49(4):41–46, 2006.

[5] P. L. T. Pirolli. *Information Foraging Theory: Adaptive Interaction with Information.* Oxford University Press, June 2004.

[6] X. Shen, B. Tan, and C. Zhai. Implicit user modeling for personalized search. In *Proceedings of CIKM 2005*, pages 824–831, 2005.

[7] J. Teevan, S. T. Dumais, and E. Horvitz. Personalizing search via automated analysis of interests and activities. In *Proceedings of ACM SIGIR 2005*, pages 449–456, 2005.

[8] X. Wang, B. Tan, A. Shakery, and C. Zhai. Search logs as information footprints: Supporting guided navigation for exploratory search. Technical Report UIUCDCS-R-2008-3001, University of Illinois, 2008. https://www.ideals.uiuc.edu/bitstream/handle/2142/10971/UIUCDCS-R-2008-3001.pdf.

[9] X. Wang and C. Zhai. Learn from web search logs to organize search results. In *Proceedings of SIGIR 2007*, pages 87–94, 2007.

[10] X. Wang and C. Zhai. Mining term association patterns from search logs for effective query reformulation. In *CIKM*, pages 479–488, 2008.

[11] R. W. White and R. A. Roth. *Exploratory Search: Beyond the Query-Response Paradigm.* Morgan and Claypool, 2009.

HCI Browser: A Tool for Studying Web Search Behavior

Robert Capra
School of Information and Library Science
University of North Carolina at Chapel Hill
rcapra3@unc.edu

ABSTRACT

We present a Mozilla Firefox extension called the HCI Browser that we are developing to support studies of how users find and refind information on the Web. The HCI Browser presents tasks to the user, collects browser event data as they search for information, records answers found, and administers pre- and post-task questionnaires.

1. INTRODUCTION

Studies of how users search, manage, and refind information on the Web often involve presenting tasks to users and observing their behaviors (e.g. web pages visited, links clicked, time spent on each page, use of the back button). Questionnaires are often administered before and after tasks to gather additional data about the participant's experiences.

Researchers have built tools such as WebTracker [5], WebLogger [4], the Curious Browser [2], and URL Tracer[1] to help support studies of web search behaviors and have noted the challenges involved with capturing naturalistic user behaviors for web search [3]. Recently, the Lemur IR toolkit project introduced the Lemur Query Log Toolbar[2], an open source browser plug-in tool that captures events such as page loads, tab switches, and searches issued to major search engines.

The tools described above are all valuable research tools, but none filled all the needs we have for collecting data on how users find and refind information on the Web. Specifically, we need a tool that will: 1) integrate with an existing Web browser to provide a familiar browsing experience, 2) record a wide variety of user interactions with the web pages and the browser itself, and 3) provide support for administrative aspects of conducting a study such as administering pre- and post- task questionnaires, recording the "answers" that participants found for the tasks given, and managing other details such as closing any opened browser windows before the start of the next tasks. To support these needs, we are developing a Mozilla Firefox extension called the HCI Browser. We are developing the HCI Browser as open-source code and have utilized some open-source code from the Lemur Query Log Toolbar project. This work also builds off our previous experience building an instrumented web browser using Visual Basic and the Microsoft Web Browser Control [1].

2. HCI BROWSER

The HCI Browser is implemented as a Firefox extension, meaning that it can easily be installed on any Firefox 3 browser. After installing the extension, every time the browser is loaded, three configuration files are read: a *task file* with the text of the tasks to present to the user, a *pre-task questions file* (Figure 1) with a set of questions to be asked prior to each task, and a *post-task questions file* with a list of questions to be asked after each task. The pre- and post-task questions can be of three different types: multiple choice, Likert-type, and free-text/open response.

When the HCI Browser is started a dialog box is shown that prompts the experimenter to enter a session number, participant number and starting task. The pre-task questions for the first task are then displayed (Figure 1). After the user clicks "OK", the main browser window is opened with the text of the task displayed in the toolbar (Figure 2). Initially, on the right side of the toolbar, buttons are provided for the user to indicate when they have found an "answer" for the task (these are not shown in Figure 2). Clicking the "found an answer" button then changes the right area of the toolbar to display textboxes for the user to enter the URL and text of the answer they have found. The URL field is pre-filled with the URL of the current page. The system can be configured to allow single or multiple answers for a task.

While the user is looking for the information on the Web, Firefox supports monitoring of a wide array of browser and user interface events including button presses, use of the history mechanism, link navigation, changes to the URL address bar, window and tab focus events, scrolling, and mouse events. Currently, the HCI Browser monitors a subset of these records them to a log file. The HCI Browser is being designed to support several modes of operation: event logging only, task presentation, and task presentation with pre- and post-questionnaires for each task. Currently, it supports pre- and post-questionnaires and limited logging. For information, downloads, and updates visit: http://ils.unc.edu/hcibrowser

UIIR Workshop SIGIR'09, July 23, 2009, Boston, USA.

[1] http://grouplab.cpsc.ucalgary.ca/cookbook/index.php/Utilities/URLTracer

[2] http://www.lemurproject.org/querylogtoolbar/

3. REFERENCES

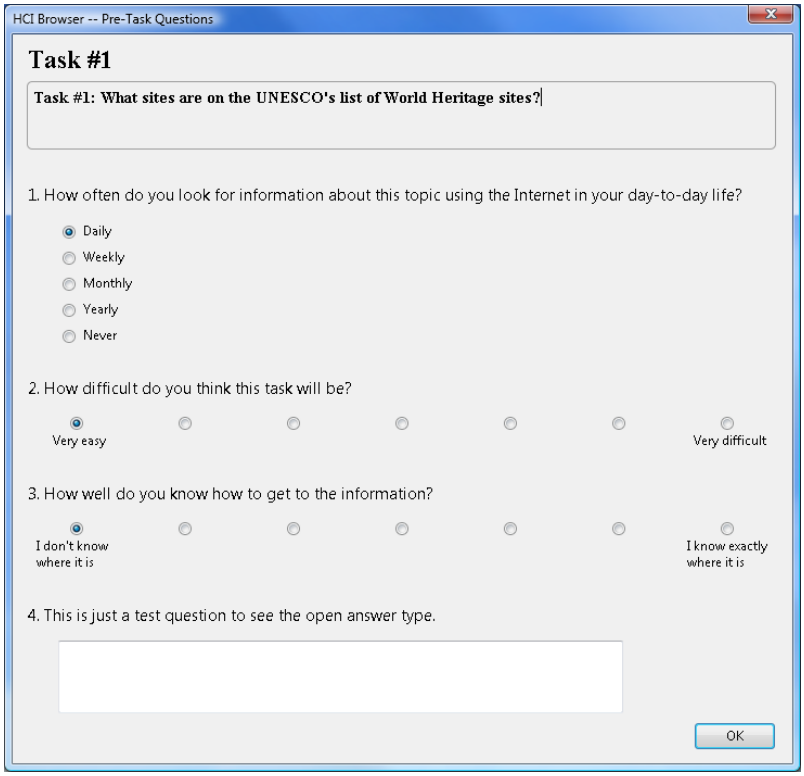
[1] Capra, R. (2008). Studying Elapsed Time and Task Factors in Re-Finding Electronic Information. Workshop on Personal Information Management at CHI 2008.

[2] Claypool, M., Le, P., Wased, M., and Brown, D. 2001. Implicit Interest Indicators. In Proceedings of the 6th International Conference on Intelligent User Interfaces.

[3] Keller, M., Hawkey, K., Inkpen, K., and Watters, C. (2008). Challenges of Capturing Natural Web-Based User Behaviors. *International Journal of Human-Computer Interaction, 24*(4), 385-409.

[4] Reeder, R. W., Pirolli, P., and Card, S. K. (2001). WebEyeMapper and WebLogger: Tools for Analyzing Eye Tracking Data Collected in Web-Use Studies. In CHI '01 Extended Abstracts.

[5] Turnbull, D. (1998). Webtacker: A Tool for Understanding Web Use. Unpublished report. Retreived on May 18, 2009 from: http://www.ischool.utexas.edu/~donturn/research/webtracker/index.html




```
Example Pre-Task Configuration File
-----------------------------------
MultipleChoice
5
How often do you look for information about
this topic using the Internet in your day-
to-day life?
Daily
Weekly
Monthly
Yearly
Never
---
LikertType
7
How difficult do you think this task will
be?
Very easy



Very difficult
---
LikertType
7
<similar to above... omitted for space>
---
OpenAnswer
This is just a test question to see the
open answer type.
```

Figure 1. HCI Browser – Pre-Task Questions

Figure 2. HCI Browser – Main Browser Window

Catching the User - Logging the Information Retrieval Dialogue

Paul Landwich
FernUniversität in Hagen
Germany
paul.landwich@fernuni-hagen.de

Claus-Peter Klas
FernUniversität in Hagen
Germany
claus-peter.klas@fernuni-hagen.de

Matthias Hemmje
FernUniversität in Hagen
Germany
matthias.hemmje@fernuni-hagen.de

ABSTRACT

This position paper supports the idea of the information dialog between IR systems and users during an information search task. In order to satisfy the communication and interaction needs of humans, IR systems should explicitly support the cognitive abilities of the users. An information dialogue which does not only support an individual query but also the complete search process is necessary. Only in this way it is possible to satisfy an information need.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Search Process; H.3.7 [**Digital Libraries**]: User Issues

General Terms

information retrieval, visualization, interactive systems

1. INTRODUCTION

Information seeking is usually not a single step to recover a piece of information, but a cyclic, highly interactive process with the aim to satisfy a specific information need. Within such a process the user builds a cognitive model, which helps her to reflect and advance the search process.Within user interfaces, it is necessary to integrate tools and functionalities within existing tools, in order to develop this cognitive perception and derive a context model of the users. Requirements for this are logging of all user and system activities ranging from entered queries to the result sets, tools to visualize the context and system support based on a context analysis.

2. ASPECTS

In order to support the statement of the introduction, we would like to dwell on three aspects.

2.1 Logging

As stated above we need to log all user and system activities and the corresponding result sets within a task to catch the users context. From the experience one knows that a search task is usually not concluded with the first query. Rather a working context through the interaction is elaborated. When this understanding becomes clear, there must be some kind of accompanying information dialogue. A dialogue consists of a sequence of activities and results.

In the past initial research ([2] and [6]) focused on the human users not only as a part of the system but also as an important component. In later works it was recognized that the search is a process. In other papers (e.g. [1]) the search strategies and search patterns where investigated. The overall complexity of the search process was exposed ([11], [12],[17]). In [7] a continuation models of information dialogue was introduced, to cover this search process.

The process of the related research was consistent: Starting from the support and improvement of individual queries, up to a more global view of the search process and dialogue. But this global view must become granular again. In order to interpret a process or a dialog, the individual steps must be identified and formalized within this dialog.

[8] identified six activities – exploration, navigation, focus, inspection, evaluation and store – to focus on to derive a context model of the user.

1. Exploration: The access to set of information objects in the form of a query and the visualization and realisation of the produced result set defines the Exploration. A change respectively an enlargement of the informal context is caused by it.

2. Focus: The focus set represents the subset of information objects of a result set which reach the field of vision of the user through a visualisation and is the result of the activity Focus

3. Navigation: The movement within a set of information objects (information room) or between different information rooms. This causes a change of the focus.

4. Inspection: Inspection is used for the cognitive determination of the state of an information object.

5. Evaluation: Evaluation gives the system a feedback of the user's understanding of relevance and appoints the verified recall set.

6. Store: This activity allows to store found documents. It either happens logically in form of a storage box on

SIGIR 2009 July 19-23, 2009, Boston, USA.

the user interface or physically when a document is downloaded or printed.

Based on these definitions we can log a dialog or the whole search process with the system. Because some of these acitivities correlate we can identify three interactive modes. The user finds oneself in one of theses modes and will change cicular the mode. The first mode is every time *access*. Within this mode there is only one activity, EXPLORATION. Already after the first EXPLORATION the user changes into the second mode *Orientation*. Activities for this mode are NAVIGATION, FOCUS and INSPECTION. The user is now in the ability to change the visual as well as the informational focal point in an information visualisation of the dialogue context. The mode *Assessment* is reached, if the user finds objects of interest during his ispection. For this mode the activities EVALUATION and STORE are available. They help to express the users appreciation of relevance and to define the identified recall set.

Beside different models for information searching ([1], [13], [5]) it was [15] who combined these approaches in a new model. Based on idea we can enhance this model with our activities and interactive modes (see figure 1).

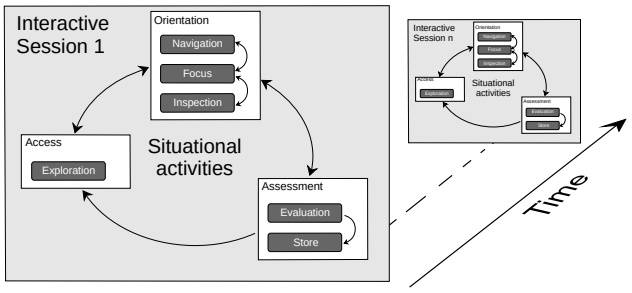


Figure 1: Enhanced model of Spink

2.2 Visualization

The past research ([14], [9]) showed that information visualization is an important concept for the cognitive support of the user. [3] said: "'Visual interfaces to IR systems exploit powerful human vision and spatial cognition to help humans mentally organize and electronically access and manage large, complex information spaces. The aim is to shift the user's mental load from slow reading to faster perceptual processes such as visual pattern recognition."'.

This statement leads us to the second aspect of our position. If we understand search as a process, whose progression fills our context, then we need also support, in order to understand and interpret this context. So the visualization of results must go beyond the usual measure. Especially the different sets of information objects shown in [7] seems to be useful to visualize (see figure 2). The user needs a portfolio of visualization tools which approach his cognitive abilities. Furthermore, the user must be able to get the full control of his search history and the developed information context. By logging all activities and the sets of information objects resulting from it, we are able to get a first formal overview of our context.

A first prototype is developed which visualize the different sets of information objects during an information search

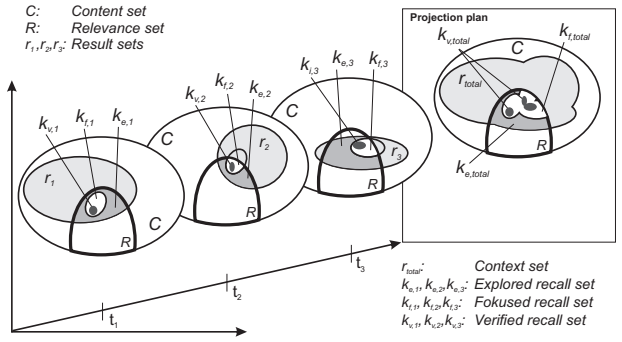


Figure 2: Sequence of Separate Queries

process (see figure 3). In a next step we will evaluate this prototype.

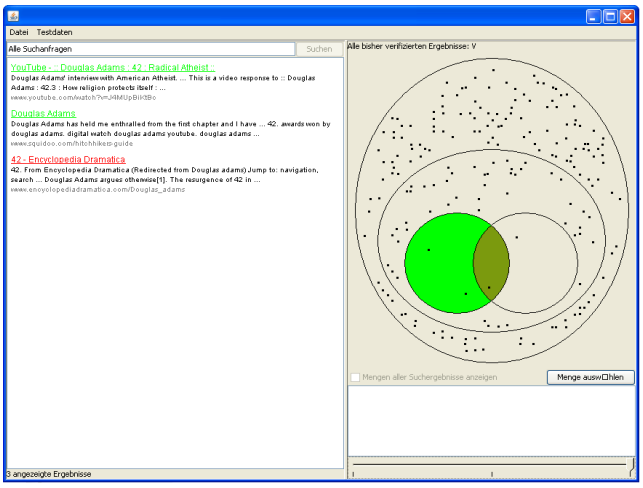


Figure 3: Screenshot of a prototype

2.3 System support

In order to support the user during the search task, systems should be proactive ([10], [16]). To be able to actually support and evaluate our model we need a system which meets the following demands. The system

- should fundamentally support the interaction model,
- should map the described activities to support the user,
- should enable the quantitative and qualitative evaluation of the model,
- and should be highly flexible and extensible to integrate new visualisation technics.

Following the formal description of the information dialogue and given the demands we want to introduce the Daffodil-system as an experimental system for further development and evaluation of the above described model. It provides already, up to a certain extend, the demand for mapping the user activities to existing available tools.

With the information of our context model including the search path we identifed the following challenges:

Relevance Feedback The users implicit and explicit relevance assessments must be captured and related to possible relevant documents.

Search strategy With the help of the user or by monitoring the activities the system must provide different search strategies to raise effciency.

Collaborative recommendations By logging many different searches in form of a set of activities, it is possible to support a user through collaborative recommendations. Analyzing a new search from the beginning, the system is able to identify similar stored search processes. If this knowledge is visualized for the user, he could get benefit for his own search task.

3. CONCLUSIONS

The idea of this position paper is to support users within a search task by logging all activities between the user and the system. For this, we are able to visualize the context and make it cognitive perceptible. Furthemore, we are able to draw conclusions from this activities. This captured information represents the basis to further understand and support the user. Such support could be done through recommendation via implicit relevance feedback as well as collaborative recommendations through other users in a similar situation. We think, that given the context model within the Daffodil-Framework, we are able to understand and categories user behavior and provide solid data to support system oriented IR evaluation, e.g. based on user simulation.

We currently investigate and evaluate our research using the Daffodil - framework ([4]) as an experimental system. In order to evaluate the listed aspects, we momentarily work on the following projects:

- Task manager: A tool to capture and log all activities and resulting sets of information objects of a search task over more then one session.
- Visualization: Visualize the context and search path with help of venn diagrams.
- Relevance feedback: Interpretation of activities as implicit relevance feedback with term suggestions and reranked result lists.

4. REFERENCES

[1] N. J. Belkin, C. Cool, A. Stein, and U. Thiel. Cases, scripts, and information-seeking strategies: On the design of interactive information retrieval systems. In *Arbeitspapiere der GMD*. GMD, Sankt Augustin, November 1994.

[2] R. . B. H. Belkin, N. J.; Oddy. Ask for information retrieval. *Journal of Documentation*, 38:61–71 (Teil 1) & 145–164 (Teil 2), 1982.

[3] K. Börner and C. Chen. Visual interfaces to digital libraries: Motivation, utilization, and socio-technical challenges. In *Visual Interfaces to Digital Libraries*, pages 1–12, London, UK, 2002. Springer-Verlag.

[4] N. Fuhr, C.-P. Klas, A. Schaefer, and P. Mutschke. Daffodil: An integrated desktop for supporting high-level search activities in federated digital libraries. In *Research and Advanced Technology for Digital Libraries. 6th European Conference, ECDL 2002*, pages 597–612, Heidelberg et al., 2002. Springer.

[5] P. Ingwersen. Cognitive perspectives of information retrieval interaction: elements of a cognitive ir theory. *Journal of Documentation*, 52:3–50, 1996.

[6] C. C. Kuhlthau. Longitudinal case studies of the information search process of users in libraries. *Library & Information Science Research*, 10:257–304, 1988.

[7] P. Landwich, T. Vogel, C.-P. Klas, and M. Hemmje. Supporting patent retrieval in the context of innovation-processes by means of information visualisation. In *Proceedings of ECKM 2008*, 2008.

[8] P. Landwich, T. Vogel, C.-P. Klas, and M. Hemmje. Model to support patent retrieval in the context of innovation-processes by means of dialogue and information visualisation. *Electronic Journal of Knowledge Management*, 7:87–98, 1 2009. http://www.ejkm.com/volume-7/v7-1/v7-i1-art9.htm.

[9] L. Nowell, E. Hetzler, and T. Tanasse. Change blindness in information visualization: A case study. In *INFOVIS '01: Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01)*, page 15, Washington, DC, USA, 2001. IEEE Computer Society.

[10] R. Oppermann. *Adaptive user support: ergonomic design of manually and automatically adaptable software*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA, 1994.

[11] N. Pharo. A new model of information behaviour based on the search situation transition schema. *Inf. Res.*, 10(1), 2004.

[12] D. E. Rose. Reconciling information-seeking behavior with search user interfaces for the web. *J. Am. Soc. Inf. Sci. Technol.*, 57(6):797–799, 2006.

[13] T. Saracevic. The stratified model of information retrieval interaction: Extension and applications. In *Proceedings of the American Society for Information Science*, volume 34, pages 313–327, 1997.

[14] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *VL '96: Proceedings of the 1996 IEEE Symposium on Visual Languages*, pages 336–343, Washington, DC, USA, 1996. IEEE Computer Society.

[15] A. Spink. A user-centered approach to evaluating human interaction with web search engines: an exploratory study. In *Information Processing and Management*, pages 401–426, 2002.

[16] M. Twidale, D. Nichols, M. B. Twidale, and D. M. Nichols. Collaborative browsing and visualisation of the search process. In *In Proceedings of ELVIRA-96, Milton Keynes*, pages 48–7, 1996.

[17] Y. Xu. The dynamics of interactive information retrieval behavior, part i: An activity theory perspective. *J. Am. Soc. Inf. Sci. Technol.*, 58(7):958–970, 2007.

Reading between the lines: identifying user behaviour between logged interactions

Max L. Wilson
Future Interaction Technologies Lab
Swansea University, UK
m.l.wilson@swansea.ac.uk

m.c. schraefel
School of Electronics and Computer Science
University of Southampton, UK
mc+uiir@ecs.soton.ac.uk

ABSTRACT
Log analyses are often used simply to quantify interactions with different aspects of a user interface. The position held here is that much of a user's search experience does not involve direct interaction with the interface, and may not be logged at all. Many models highlight the cognitive aspects of searching behaviour, and many consider that if a user does not like a user interface, then they do not interact with it very much. Consequently, we suggest that a grand challenge for logging searcher experiences should be to study the gaps in usage logs rather than the entries alone.

INTRODUCTION
Searching involves both mental and physical actions [1, 3, 4, 6, 8-10, 16]. Whether a user is reading, scanning, choosing, or thinking of query terms, there are many agreed elements of the search process, or search experience, which do not involve interacting directly with the computer. The problem with logging user interactions, therefore, is that it provides only half of the picture. When a user finds it hard to use a search interface, they may not find it hard to click or type, but instead find it hard to work out what to do first, where to go next, or why something happened. The issue is further highlighted when we consider interface features that are primarily for orientation or feedback, like breadcrumbs.

The think aloud approach is one example method used for eliciting qualitative details of user experience, but both the experimenter effect and the weaknesses of introspection are well known [14]. Some physiological logging approaches, such as eye tracking, heart rate, body temperature, and pupil-size monitoring can also be used if the participant is in a lab environment. Studies even consider brain scanning methods to estimate user cognitive load [5]. Can we elicit cognitive aspects from logs of distant users? This position paper explores the potential of eliciting cognitive actions from usage logs, which we know are part of search.

COGNITIVE ACTIONS DURING SEARCH
Many models of information seeking assume that there are cognitive stages in the search process. Marchionini [10], Ellis [4], and Kuhlthau [9], all identify stages such as need identification, examining results, and reflecting on whether a task has been completed. Similarly relevance judgments [11] are presumed to be a key part of searching as a user chooses which search results to view.

Many analytical evaluation methods for interfaces define cognitive actions. The Keystroke Level Model (KLM) [3] was designed to estimate how long it would take to perform a task with a user interface, by providing time estimates for actions like clicking and typing. Further, KLM suggests that the average time for a mental action is around 1.2 seconds and may include actions such as: initiating a task, making a strategy decision, retrieving a chunk from memory, visual search on the screen, thinking of a task parameter (like a keyword for a query), and verifying that something has happened. The GOMS method (Goals, Operators, Methods, and Selection rules) identified two types of non-interactive actions: cognitive and perceptual. Cognitive actions include initiating, choosing, planning. Perceptual actions include reading and performing visual search. These were later made more explicit in a variation called CPM-GOMS (Cognitive-Perceptual-Motor GOMS – also Critical Path Method GOMS), suggesting these cognitive, perceptual and motor (interactive) actions may occur in parallel [7].

Bates discussed both mental and physical actions in an analysis of different levels of search strategies [1]. Her model, which was operationalised in a recent information seeking evaluation framework [16], suggests that there are four levels of strategy: Strategies, Stratagems, Tactics, and Moves. She defines these moves as 'An identifiable thought or action that is a part of information searching'. Tactics, such as comparing, narrowing results, expanding results, varying queries, etc, are made up of moves. Stratagems, such as checking journal issues or searching for citations, are made up of a combination of tactics and joining moves. Finally strategies, which are similar to realistic work tasks like verifying a citation, or researching for a report, are made up of a combination of stratagems, tactics, and moves. Consequently, all four levels involve cognitive actions. Bates' definition of moves, and subsequently the information seeking evaluation method by Wilson and colleagues, takes a much less rigid view of mental actions compared to timeframe analyses like KLM and GOMS.

INTERFACE ELEMENTS FOR FEEDBACK
Elements or features of user interfaces are often designed to provide feedback to users or support orientation. Although these often-passive elements, like breadcrumbs, *can* be used to navigate around an interface, they may be often used

without any direct interaction. Anecdotally, Pickens has blogged about the dependence on usage logs[1] and the value that can be gained from classifications without direct interaction[2]. This topic was discussed in the CHI09 Sensemaking workshop. Further, at CHI09, an audience question asked whether tag clouds are better for aiding retrieval or providing contextual information about results. Empirically, Wilson and colleagues have shown that users can recall labels from faceted classifications that did not receive direct interaction [15].

IDENTIFYING COGNITIVE ACTIONS IN USAGE LOGS

The solution for identifying cognitive actions from usage logs is by no means obvious. Several existing studies, however, can provide some insights into how we might begin to do so. Multiple studies have, for example, noted that users sometimes move their mouse to the most relevant result seen so far while continuing to scan results [2, 12]. The combination of eye tracking and mouse tracking used tells us more about both perceptual actions (scanning the results) and cognitive actions (judging relevance), before interaction occurs (clicking). Further this reinforces the notion that we can use triangulation of, in this case, logging methods to build richer pictures of search experiences.

Similarly, in a study performed by schraefel and colleagues [13], audio previews were provided with labels in the facets of a classical music dataset. The hypothesis was that multiple previews would improve user choices while browsing, and would 'back out' of their decisions less often. This mental action of 'backing out' on a decision was measured in logs by a pattern of interactions showing the user clicking on higher levels of the classifications from their previous position. In this case, therefore, certain cognitive actions were modeled as a sequence of physical interactions, in an environment where mouse and eye tracking were not used. Although schraefel and colleagues identified specific mental actions, it may be possible to identify common interaction patterns that abstractly represent known perceptual and cognitive search Moves.

CONCLUSIONS

Search is irrefutably made up of both mental and physical actions: we cannot interact with a system without first choosing how to interact with it. The challenge, therefore, is to try to elicit common mental actions from logs of physical interactions. There are two key avenues that we envisage for beginning to do so. First, triangulation of multiple measures is already known to provide a richer understanding of user experiences and applies to logging too. Second, modeling sequences of physical interactions may allow us to estimate what has happened in the gaps. Regardless of how it is eventually achieved, the key position held here is that evaluating searcher experiences with usage logs should focus on what happens between the captured physical interactions.

[1] http://irgupf.com/2009/05/26/machine-learning-and-search-action-or-reaction/

[2] http://thenoisychannel.com/2009/03/24/google-offers-more-and-better-search-refinements/

REFERENCES

[1] Bates, M.J. Where should the person stop and the information search interface start? Inf. Process. Manage., 26, 5 (1990). 575-591.

[2] Brumby, D.P. and Howes, A. Strategies for Guiding Interactive Search: An Empirical Investigation Into the Consequences of Label Relevance for Assessment and Selection. Hum.-Comput. Interact., 23, 1 (2008). 1-46.

[3] Card, S.K., Moran, T.P. and Newell, A. The keystroke-level model for user performance time with interactive systems. Commun. ACM, 23, 7 (1980). 396-410.

[4] Ellis, D. A behavioural approach to information retrieval system design. J. Doc., 45, 3 (1989). 171-212.

[5] Hirshfield, L., et al., Brain measurement for usability testing and adaptive interfaces: an example of uncovering syntactic workload with functional near infrared spectroscopy. in CHI'09, (2009), 2185-2194.

[6] Ingwersen, P. Cognitive Information Retrieval. ARIST, 34 (1999). 3-52.

[7] John, B. and Gray, W., CPM-GOMS: an analysis method for tasks with parallel activities. in CHI'95, (1995), ACM, 393-394.

[8] John, B. and Kieras, D. The GOMS family of user interface analysis techniques: Comparison and contrast. ACM Trans. Comput.-Hum. Interact., 3, 4 (1996). 320-351.

[9] Kuhlthau, C.C. Inside the search process: Information seeking from the user's perspective. JASIS, 42, 5 (1991). 361-371.

[10] Marchionini, G. Information Seeking in Electronic Environments. Cambridge University Press, 1995.

[11] Rocchio, J. Relevance feedback in information retrieval. in Salton, G. ed. *The SMART Retrieval System: Experiments in Automatic Document Processing*, Prentice Hall, 1971, 313-323.

[12] Rodden, K. and Fu, X., Exploring how mouse movements relate to eye movements on web search results pages. in WISI'07, (2007).

[13] schraefel, m.c., Wilson, M.L. and Karam, M. Preview Cues: Enhancing Access to Multimedia Content, School of Electronics and Computer Science, University of Southampton, 2004.

[14] Van Someren, M., Barnard, Y. and Sandberg, J. The Think Aloud Method: A practical guide to modelling cognitive processes. Academic Press London, 1994.

[15] Wilson, M.L., André, P. and schraefel, m.c., Backward Highlighting: Enhancing Faceted Search. in UIST'08, (2008), ACM, 235-238.

[16] Wilson, M.L., schraefel, m.c. and White, R.W. Evaluating Advanced Search Interfaces using Established Information-Seeking Models. JASIST (2009).