# HCIR 2009

Proceedings of the Third

## Workshop on Human-Computer Interaction and Information Retrieval

http://cuaslis.org/hcir2009

The Catholic University of America, Washington DC, USA

October 23, 2009

Workshop Chairs:

Bill Kules, The Catholic University of America
Daniel Tunkelang, Endeca
Ryen White, Microsoft Research

Supporters:

THE CATHOLIC UNIVERSITY of AMERICA

ENDECA®

Microsoft® Research

# Third Workshop on
# Human-Computer Interaction and Information Retrieval

When we held the first HCIR workshop in 2007, the idea of uniting the fields of Human-Computer Interaction (HCI) and Information Retrieval (IR) was a battle cry to move this research area from the fringes of computer science into the mainstream. Two years later, as we organize this third HCIR workshop on the heels of a highly successful HCIR 2008, we see some of the fruits of our labor. Topics like interactive information retrieval and exploratory search are receiving increasing attention, among both academic researchers and industry practitioners.

But we have only begun this journey. Most of the work in these two fields still stays within their silos, and the efforts to realize more sophisticated models, tools, and evaluation metrics for information seeking are still in their early stages.

In this year's one-day workshop, we will continue to explore the advances each domain can bring to the other.

# Table of Contents

## Panel Papers

## Poster Papers

# Usefulness as the Criterion for Evaluation of Interactive Information Retrieval

M. Cole, J. Liu, N. J. Belkin, R. Bierig, J. Gwizdka, C. Liu, J. Zhang, X. Zhang
School of Communication and Information
Rutgers University
4 Huntington Street, New Brunswick, NJ 08901, USA
{m.cole, belkin, bierig, jacekg}@rutgers.edu, {jingjing, changl, zhangj}@eden.rutgers.edu, xiangminz@gmail.com

## ABSTRACT

The purpose of an information retrieval (IR) system is to help users accomplish a task. IR system evaluation should consider both task success and the value of support given over the entire information seeking episode. Relevance-based measurements fail to address these requirements. In this paper, *usefulness* is proposed as a basis for IR evaluation.

## Categories and Subject Descriptors

H.1.2 [**Models and Principles**]: User/Machine Systems – *human information processing* H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *search process*

## General Terms

Measurement, Performance, Experimentation, Human Factors

## Keywords

Evaluation, Information seeking, Interaction, Usefulness

## 1 INTRODUCTION

Research in information retrieval (IR) has expanded to take a broader perspective of the information seeking process to explicitly include users, tasks, and contexts in a dynamic setting rather than treating information search as static or as a sequence of unrelated events. The traditional Cranfield/TREC IR system evaluation paradigm, using document relevance as a criterion, and evaluating single search results, is not appropriate for interactive information retrieval (IIR). Several alternatives to relevance have been proposed, including utility and satisfaction. We have suggested an evaluation model and methodology grounded in the nature of information seeking and centered on *usefulness* [1] [2]. We believe this model has broad applicability in current IR research. This paper extends and elaborates the model to provide grounding for practical implementation.

## 2 INFORMATION SEEKING

As phenomenological sociologists (e.g., [7]) note, people have their life-plans and their knowledge accumulates during the process of accomplishing their plans (or achieving their goals). When personal knowledge is insufficient to deal with a new experience, or to achieve a particular goal, a *problematic situation* arises for the individual and they seek information to resolve the problem [7]. Simply put, information seeking takes place in the circumstance of having some goal to achieve or task to complete.

We can then think of IR as an information seeking episode consisting of a sequence of interactions between the user and information objects [4]. Each interaction has an immediate goal, as well as a goal with respect to accomplishing the overall goal/task. Each interaction can itself be construed as a sequence of specific *information seeking strategies* (ISSs) [8].

We believe appropriate evaluation criteria for IR systems are determined by the system goal. The goal of IR systems is to support users in accomplishing the task/achieving the goal that led them to engage in information seeking. Therefore, IR evaluation should be modeled under the goal of information seeking and should measure a system's performance in fulfilling users' goals through its support of information seeking.

## 3 GOAL, TASK, SUB-GOAL & ISS

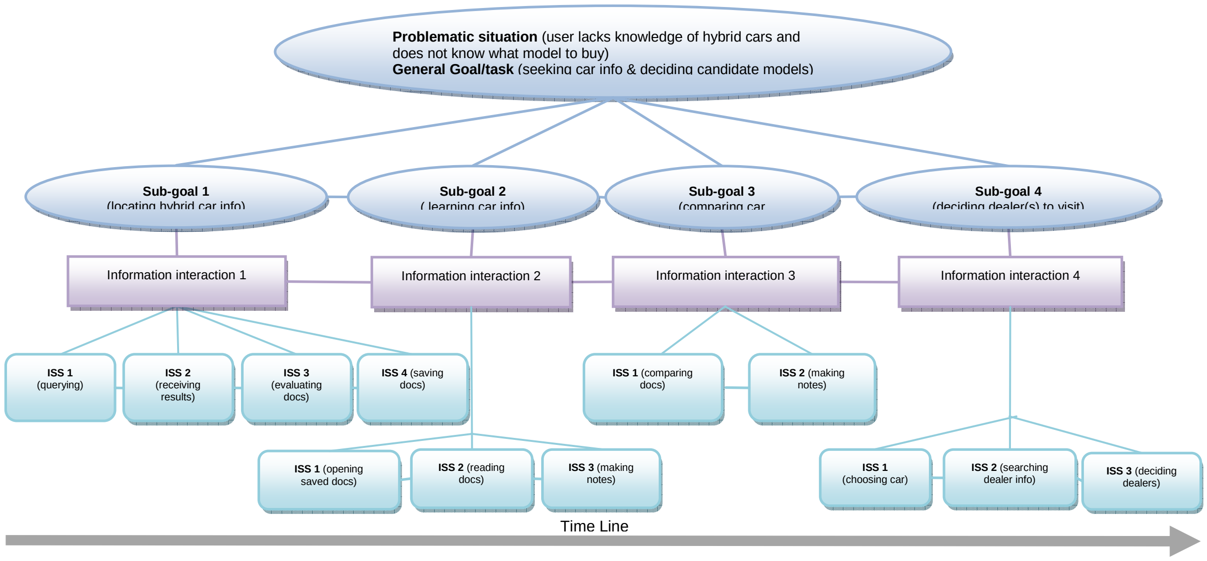
In accomplishing the general work task and achieving the general goal, a person engaged in information seeking goes through a sequence of information interactions (which are sub-tasks), each having its own short term goal that contributes to achieving the general goal. Figure 1 illustrates the relationships between the task/goal, sub-task/goal, information interaction, and an ISS.

Let us give an example. Suppose someone in need of a hybrid car wants to choose several car models as candidates for further inspection at local dealers. The *problematic situation* [7] is that he lacks knowledge on hybrid cars. His general *work task* is seeking hybrid car information and deciding at which models he should look. He may go through a sequence of steps which have their own *short-term goals*: 1) locating hybrid car information, 2) learning hybrid car information, 3) comparing several car models, and 4) deciding which local dealers to visit. In each *information interaction* with a short-term goal, he may go through a sequence of *ISSs*. For example, searching for hybrid car information may consist of querying, receiving search results, evaluating search results, and saving some of them.

There are several general comments. First, Figure 1 shows only the simplest linear relations between the steps along the time line. In fact, the sequence of steps/sub-goals/ISSs could be non-linear. For instance, on the sub-goal level, after learning hybrid car information, the user may go back to an interaction of searching for more information. At the ISS level, after receiving search results, the user may go back to the querying step.

Second, the contribution of each sub-goal to the general goal may change over time. For instance, suppose in one information interaction, the user looks at information of car model 1 and decides to choose it as a final candidate. After he learns about car model 2, which outperforms car model 1 in all aspects, he removes model 1 from the candidate list. Therefore, some steps in the sequence (choosing car model 1) may contribute to the

HCIR Workshop 2009, Washington, D.C., October 23, 2009.


**Problematic situation** (user lacks knowledge of hybrid cars and
does not know what model to buy)
**General Goal/task** (seeking car info & deciding candidate models)
**Sub-goal 1**
(locating hybrid car info)
**Sub-goal 2**
( learning car info)
**Sub-goal 3**
(comparing car
**Sub-goal 4**
(deciding dealer(s) to visit)
Information interaction 1
Information interaction 2
Information interaction 3
Information interaction 4
**ISS 1** (querying)
**ISS 2** (receiving results)
**ISS 3** (evaluating docs)
**ISS 4** (saving docs)
**ISS 1** (opening saved docs)
**ISS 2** (reading docs)
**ISS 3** (making notes)
**ISS 1** (comparing docs)
**ISS 2** (making notes)
**ISS 1** (choosing car)
**ISS 2** (searching dealer info)
**ISS 3** (deciding dealers)
Time Line

**Evaluation based on the following three levels:**

*1. The usefulness of the entire information seeking episode with respect to accomplishment of the leading task;*
*2. The usefulness of each interaction with respect to its contribution to the accomplishment of the leading task;*
*3. The usefulness of system support toward the goal(s) of each interaction, and of each ISS*

**Figure 1. An IIR Evaluation Model**

sub-goal positively, but it contributes to the final and overall goal negatively in that car model 1 is eventually removed.

Third, the leading goal of this task is, or can be taken to be, relatively stable over the course of the interaction. Different users can and will do different things to achieve similar leading goals. Some of the differences in these sequences may be characteristics of classes of users, for example, high/low domain knowledge, cognitive capacities, and of task types, including task complexity.

# 4 AN EVALUATION MODEL

Fundamentally, we are interested in why a person engages an information need and how an interaction session contributes to meeting that need. It follows one must provide a measurement for the session as a whole and for the session constituents.

## 4.1 Three levels of evaluation

A user makes progress towards a goal by virtue of the results of interactions with the system. Support of results and process are two aspects of system performance. Evaluation of a system should center on how well the user is able to achieve their goal, the process of helping the user identify and engage in appropriate interactions, and the relationship of the results of those interactions to the progress toward and accomplishment of the goal. IR evaluation should then be conducted on three levels. First, it should evaluate the information seeking episode as a whole with respect to the accomplishment of the user's task/goal. Second, it should assess each interaction, meaning explicitly the effectiveness of support for each ISS, with respect to its immediate goal. Third, it should assess each interaction with respect to its contribution to the accomplishment of the overall task/goal.

An ideal system will support its users' task accomplishment by presenting resources and user support in an optimally-ordered minimum number of interaction steps (cf. [3]). Resources and user support should address not only search result content, i.e., techniques to rank the most relevant documents at the top, but they should also be manifest in the system interface, including search interface, result display, and various ways to support general task accomplishments. For example, the system could have a function of comparing pages that users have seen for them to better understand or summarize what they have learned about the task topic, so as to help them in solving the task. As another example, the system may have a place for the users to make notes, or create document drafts, which on one hand, is a way of helping users start generating their task-solving documents, and, on the other hand, are helpful for relevance feedback/query reformulation.

## 4.2 Criterion: Usefulness

We propose *usefulness* as the criterion for IIR evaluation. Existing measures of IR performance are inadequate for the proposed IIR evaluation model.

The sense of usefulness we have in mind is more general than relevance, which has come, for historical reasons [1], to be the received basis for measuring IR systems. Like relevance, people are able to give usefulness judgments as intuitive assessments that do not turn on understanding a technical definition. Usefulness, however, is suited to interaction measurements in ways relevance-based systems cannot address.

The problem of measuring IIR has recently received attention in terms of formal models (e.g. [4]) and the relation of local interactions to realization of search session outcomes (e.g. [5]). Usefulness measurements are distinguished from session-level measurements like Järvelin, et al.'s session-based discounted cumulative gain (sDCG) [5] in that usefulness explicitly considers the session as a whole. sDCG does not support judgment of relevance to the whole session or how results from an interaction step might be integrated into the whole. It depends on the assumption the only thing that matters is the relevance of the local interaction and the incremental change it makes on the history of relevance judgments to that point.

Usefulness is specifically distinguishable from relevance in several dimensions. Most strikingly, a usefulness judgment can be explicitly related to the perceived contribution of the judged object or process to progress towards satisfying the leading goal or a goal on the way. In contrast to relevance, a judgment of usefulness can be made of a result or a process, rather than only to the content of an information object. It also applies to all scales of an interaction. Usefulness can be applied to a specific result, to interaction subsequences, and to the session as a whole. Usefulness, then, is more general than relevance, and well-suited to the object of providing a measurement appropriate to the concept of task goal realization.

This does not deny the importance of relevance as a specific measurement to be used in appropriate circumstances to determine usefulness. For example, relevance can be used as a usefulness criterion for interaction steps where the immediate goal is to gather topical documents. Here, it is the aboutness of a document that constitutes its usefulness to advancing the task, so relevance is the appropriate usefulness criterion. This example illustrates a larger point tied with the generality of usefulness as a measure. Measuring usefulness relies on adopting appropriate and varied criteria, even within a task session. Examples of such criteria include explicit judgments including relevance and usefulness, and implicit markers, such as decision and dwell times on documents, number of steps to complete a sub-goal, user's actions to save, revisit, classify and use documents, and issue and reformulate queries. Researchers already use specific criteria such as these for evaluation. One consequence of adopting usefulness is that several measures should be used, and perhaps only for specific segments in the episode. Identifying which measures are important for episode components and for the entire episode must be experimentally determined.

Usefulness should be applied both for the entire episode against the leading goal/task and, independently, for each sub-task/interaction in the episode. Specifically, 1) How useful is the information seeking episode in accomplishing the leading task/goal? 2) How useful is each interaction in helping accomplish the leading task? 3) How well was the goal of the specific interaction accomplished? From the system perspective, evaluation should focus on: 1) How well does the system support the accomplishment of the overall task/goal? 2) How well does the system support the contribution of each interaction towards the achievement of the overall goal? 3) How well does the system support each interaction?

## 4.3 Measurements

Identifying specific measures of usefulness and how to obtain them are clearly difficult problems. The most important aspect of this evaluation framework is that it depends crucially upon specification of a leading task or goal whose accomplishment can itself be measured.

Generally, operationalization of usefulness at the level of the IR episode will be specific to the user's task/goal; at the level of contribution to the outcome it will be specific to the empirical relationship between each interaction and the search outcome; and finally, at the third level, it will be specific to the goals of each interaction/ISS.

Examples at each level might be: the perceived usefulness of the located documents in helping accomplish the whole task; task accomplishment itself, in terms of correctness, effort, or time; the extent to which systems suggestions as to what to do are taken up; the extent to which documents seen in an interaction are used in the solution; the degree to which useful documents appear at the top of a results list; and the extent to which suggested query terms are used, and are useful.

As an example, consider the hybrid car information seeking episode and focus on just the leading goal/task, sub-goal 1 and the information interaction 1 with its four ISSs. To demonstrate how the criterion of usefulness can be operationalized, the evaluation could be approached from the following aspects (this is not intended as an exclusive list):

- **at the level of the whole episode [leading goal/task]**
  - *accomplishment of the task [result]*
    - How well did the user successfully select candidate car models? [correctness]
    - How many steps (e.g., interactions, ISSs) did the user go through for the whole task? [effort]
    - How long did the user spend to complete the whole task? [time]
  - *support to the information seeking episode [process]*
    - How useful was the system in supporting identification of appropriate sub-goals in selecting hybrid car models?
    - Were system suggestions on what to do (e.g., a system suggesting four task steps: locating information, learning, comparing, and deciding) accepted?
    - How well did the system support the user in choosing an appropriate sub-task sequence?
- **at the level of the information interaction/sub-goal [sub-goal 1/information interaction 1]**
  - *accomplishment of the sub-goal [result]*
    - How well did the user successfully locate hybrid car information? [correctness]
    - How many steps (e.g., ISSs) did the user go through in locating car information? [effort]
    - How long did the user spend to locate car information? [time]
  - *support to information interaction 1[process]*
    - How useful was the system in supporting users to identify appropriate ISSs in locating hybrid car information?
    - Were system suggestions on what to do (e.g., suggesting a user should now query, view results, evaluate results or save documents) accepted?
    - How well did the system support the user in choosing an appropriate ISS sequence?
- **at the level of the contribution of the sub-goal to the leading goal [sub-goal 1 to the leading goal]**
  - *accomplishment of the contribution [result]*
    - How much did locating car information contribute to the whole task of selecting candidate car models?
  - *support to this contribution [process]*
    - How useful was the system in supporting users to locate car information in order to finally select candidate car models?

- **at the level of the ISSs [ISSs 1-4 information interaction 1]**
  - o How useful were suggested queries/terms for formulating queries? [ISS1]
  - o How much were the suggested queries/terms used? [ISS1]
  - o How well does the system support evaluation of retrieved documents? [ISS3]
  - o How well does the system support saving or retaining the retrieved, or useful, documents? [ISS4]
- **at the level of the contribution of each ISS to the sub-goal or leading goal [ISSs 1-4 to sub-goal 1 and leading goal]**
  - o How useful were the suggested queries/terms (for systems with query formulation assistance) for locating car information? [ISS1 to sub-goal 1]
  - o How well did the system rank documents? (using relevance and various other measures: precision, DCG, etc.) [ISS2 to sub-goal 1]
  - o How useful was each viewed document in helping users locate hybrid car information? [ISS2 to sub-goal 1]
  - o How useful was each viewed document in helping users select the candidate car models? [ISS2 to leading goal]

### 4.4 Experimental frameworks for IIR system measurement

One challenge in measuring the performance of IIR systems is to move beyond the Cranfield and TREC relevance-based models. Several experimental frameworks are available to measure system performance over interactive sessions.

Traditional user-studies can be used by setting a task with a measurable outcome that is related to information seeking activities. Systems are then compared by both outcome and the interaction path taken to task completion. Our proposal addresses how the interaction path can be assessed to measure its contribution to the outcome.

The limitations of user-studies are scale-related. One can address only a small number of tasks with a limited number of subjects. User-studies have the virtue of well-specified tasks and the ability to collect many details about users and their interactions.

An alternative framework, in the spirit of A-B system comparisons often used in commercial settings, is to make available two versions of a system and compare measures as people make use of the system (e.g. [6]). A big advantage of this approach is the ability to conduct large-scale tests with many users and (implicitly) many tasks. The limitation is that one needs to infer properties of the tasks and also the usefulness of the system response to meeting the needs of the users. One difficult technical issue is the identification of sessions to enable session-level results analysis.

A third, somewhat intermediate, approach to achieve reasonable scale with enough detail to enable a rich assessment of system performance for user task support, is to build a reference database of session interactions. This might be assembled in a cooperative effort and made available to research groups to generate system performance results. Such a usefulness-based interaction database would presumably include user models to choose interaction outcomes depending on the choices offered by the system and the support provided by the system at each step along the way. Such a database might be generated from uniformly-instrumented user studies and a reference user model(s).

## 5 CONCLUSION

Information retrieval is an inherently and unavoidably interactive process, which takes place when a person faces a problematic situation with respect to some goal or task. Thus, evaluating IR systems must mean both evaluating their support with respect to task accomplishment, and evaluating them with respect to the entire information seeking episode. Past, and most current approaches to IR evaluation, as exemplified by TREC, fail to address either of these desiderata, focusing as they do on relevance as the fundamental criterion, and on effectiveness of system response to a single query. In this paper, we propose an alternative evaluation model which attempts to address both of these issues, based on the criterion of *usefulness* as the basis for IR evaluation. Although our proposed model clearly needs more detailed explication, we believe that it offers a useful basis from which realistic and effective measures and methods of IR evaluation can be developed.

## 6 ACKNOWLEDGMENTS

These ideas have benefited from discussion at the 2009 Dagstuhl Seminar on Interactive Information Retrieval, especially the contributions of Pertti Vakkari, Kal Järvelin, and Norbert Fuhr, An earlier version of this paper was presented at the SIGIR 2009 Workshop on The Future of IR Evaluation, and discussion there has substantially influenced this version. This work is supported by IMLS grant LG-06-07-0105-07.

## 7 REFERENCES

[1] Belkin, N.J, Cole, M., and Bierig, R. (2008). Is relevance the right criterion for evaluating interactive information retrieval? In *Proceedings of the SIGIR 2008 Workshop on Beyond Binary Relevance: Preferences, Diversity, and Set-Level judgments*, (Singapore, 2008). Retrieved from: http://research.microsoft.com/en-us/um/people/pauben/bbr-workshop/talks/belkin-bbr-sigir08.pdf on August 24, 2009.

[2] Belkin, N.J., Cole, M. and Liu, J. (2009). A model for evaluation of interactive information retrieval. In *Proceedings of the SIGIR 2009 Workshop on the Future of IR Evaluation* (Boston). IR Publications, Amsterdam. 7-8.

[3] Belkin, N.J., Cool, C., Stein, A. and Thiel, U. (1995). Cases, scripts, and information-seeking strategies: On the design of interactive information retrieval systems. *Expert Systems with Applications*, 9(30). 379-395.

[4] Fuhr, N. (2008). A probability ranking principle for interactive information retrieval. *Information Retrieval*, 11.251-265.

[5] Järvelin, K., Price, S.L., Delcambre, L.M.L., and Nielsen, M.L. (2008). Discounted cumulative gain based evaluation of multiple-query IR sessions. In *Proceedings of the 30th European Conference on Information Retrieval* (Glascow, Scotland, 2008), Springer-Verlag. 4-15.

[6] Joachims, T. (2002). Optimizing search engines using clickthrough data. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Edmonton, Canada, 2002) ACM. 133-142.

[7] Schutz, A. and Luckmann, T. (1973). *The structures of the life-world.* Northwestern University Press, Evanston, IL.

[8] Yuan, X.-J. and Belkin, N.J. (2008). Supporting multiple information-seeking strategies in a single system framework. In *Proceedings of the 31st ACM SIGIR International Conference on Research and Development in Information Retrieval* (Singapore, 2008). ACM. 247-254.

# Modeling Searcher Frustration

Henry Feild and James Allan

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

## ABSTRACT

When search engine users have trouble finding what they are looking for, they become frustrated. In a pilot study, we found that 36% of queries submitted end with users being moderately to extremely frustrated. By modeling searcher frustration, search engines can predict the current state of user frustration, tailor the search experience to help the user find what they are looking for, and avert them from switching to another search engine. Among other observations, we found that across the fifteen users and six tasks in our study, frustration follows a law of conservation: a frustrated user tends to stay frustrated and a non-frustrated user tends to stay not frustrated. Future work includes extracting features from the query log data and readings from three physical sensors collected during the study to predict searcher frustration.

## Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*relevance feedback, search process*

## General Terms

Human Factors, Measurement

## Keywords

Information retrieval, human-computer interaction, frustration modeling

## 1. INTRODUCTION

In this work, we investigate modeling *searcher frustration.* We consider a user to be frustrated when their search process is impeded, regardless of the reason. A frustration model capable of predicting how frustrated searchers are throughout their search is useful retrospectively to collect statistics about the effectiveness of a search system. More importantly, it allows for real-time system intervention of frustrated searchers, hopefully preventing users from leaving for another search engine or just giving up. Evidence from users' interactions with the search engine during a task can be used to predict a user's level of frustration. Depending on the level of frustration and some classification of the *type* of frustration, the system can change the underlying retrieval algorithm or the actual interface. For example, we posit that one common cause or type of frustration is a user's inability to formulate a query for their otherwise well defined information need.

One way that a system could adapt to address this kind of frustration is to show the user a conceptual break down of the results; rather than listing all results, group them based on the key concepts that best represent them. So if a user enters 'java', they can see the results based on 'islands', 'programming languages', 'coffee', etc. Of course, most search engines already strive to diversify result sets, so documents relating to all of these different facets of 'java' are present, but they might not stick out to some users, causing them to become frustrated.

An example from the information retrieval (IR) literature of a system that adapts based on a user model is work by White, Jose, and Ruthven [5]. They used implicit relevance feedback to detect the changes in the type of information need of the user and alter the retrieval strategy. In our work, we want to detect frustrated behavior and adapt the system based on the type of frustration.

While automatic frustration modeling has not been specifically investigated in the IR literature, it has been explored in the area of intelligent tutoring systems (ITS) research. When a system is tutoring a student, it is helpful to track that student's affective state, including frustration, in order to adapt the tutoring process to engage the student as much as possible. Our research borrows heavily from the tools used in and insights gleaned from the ITS literature.

The goals for our line of research are as follows: first, determine how to detect a user's level of frustration; second, determine what the key causes or types of frustration are; and third, determine the kinds of system interventions that can counteract each type of frustration. Our current work focuses on the first two, leaving the third for future studies.

## 2. RELATED WORK

Our research is based heavily on two bodies of work: one from the IR literature and the other from the ITS literature. We will first describe work by Fox et al. [3] in IR followed by the work of Cooper et al. [2] and Kapoor, Burleson, and Picard [4] in the field of ITS.

## 2.1 Predicting searcher satisfaction

Fox et al. [3] conducted a study to determine if there is an association between implicit measures of user interest derived from query logs and explicit user satisfaction. They collected satisfaction feedback for every non-search engine page visited and for every session (see Section 3 for the definition of session).

Fox et al. found there exists an association between query log features and searcher satisfaction, with the most predictive features being click through, the time spent on the search result page, and the manner in which a user ended a search. Using a Bayesian model, they were able to predict the level of satisfaction with 57% accuracy at the results level (with a baseline of 40%) and 70% at the session-level (with a baseline of 56%).

In our work, we extend the research of Fox et al. to include a satisfaction feedback prompt for every individual query. We also ask users to rate their frustration with the search process and the degree to which each query's results meet their expectations. In addition, our scales are finer—five levels for all feedback rather than three—which should allow users to better assess themselves. Our work in part explores if the success of modeling user satisfaction with query log features transfers to modeling frustration.

## 2.2 Detecting ITS user emotion

Cooper et al. [2] describe a study in which students using an intelligent tutoring system were outfitted with four sensors: a mental state camera that focused on the student's face, a skin conductance bracelet, a pressure sensitive mouse, and a chair seat capable of detecting posture. The goal of the study was to ascertain if using features drawn from the sensor readings in combination with features extracted from user interaction logs with the ITS could more accurately model the user's affective state than using the interaction logs alone.

Cooper et al. found that across the three experiments they conducted, the mental state camera was the best stand-alone sensor to use in conjunction with the tutoring interaction logs for determining frustration. However, using features from all sensors and the interaction logs performed best. They used step-wise regression to develop a model for describing each emotion. For frustration, the most significant features where from the interaction logs and the camera, though features from all sensors were considered in the regression. The model obtained an accuracy of 89.7%; the baseline—guessing that the emotional state is always low—resulted in an accuracy of 85.29%.

In a related study using the same sensors, but different features, Kapoor, Burleson, and Picard [4] created a model to classify ITS user frustration. They achieved a classification accuracy of 79% with a chance accuracy of 58%. These studies demonstrate the utility of the sensor systems for predicting ITS user frustration. In our research, we will explore how well these sensors predict searcher frustration.

# 3. DEFINITIONS

To be clear, we will use the following definitions.

**Task.** A task is a formal description of an information need.

**Query.** A query is the text submitted to a search engine. We also discuss query level events, which refer to the user interactions with the results returned for the query and any subsequent pages visited until the next query is entered or the session is completed, whichever comes first.

**Session.** A session consists of all of the user interactions while searching for information for a particular task.

**Satisfaction.** We define satisfaction as the fulfillment of a need or want; in the case of IR, a user's information need. For example, a user can be asked the degree to which a Web page, query, or session fulfilled their information need.

**Frustration.** We consider a user frustrated when their search process is impeded, regardless of the reason. To measure frustration, we ask users to rate their level of frustration with the current task up to the current point on a scale of 1 (not frustrated at all) to 5 (extremely frustrated). A user is considered frustrated if they indicate a level of 3 or more. While satisfaction and frustration are closely related, they are distinct. As a consequence, a searcher can ultimately satisfy their information need (i.e., be satisfied), but still have been quite frustrated in the process [1].

# 4. USER STUDY

We conducted a user study consisting of 15 undergraduate and graduate students, each of which was asked to find information for the same six tasks using the Web. Their interactions with the browser were logged along with data from three physical sensors. The subjects were asked to assess their level of satisfaction at the result, query, and session levels, their frustration at the query level, and the degree to which the results returned for each query met their expectations. We describe each of the aspects of the study in more detail below.

## 4.1 Tasks

Subjects were asked to search for information to satisfy six tasks. Here we give a brief description of each along with a label in italics at the beginning of the description.

- *[Thailand]* Search the Web to make a list of pros and cons of a trip to Thailand.
- *[Anthropology]* Search the Web for decent anthropology programs that are as close to Ohio as possible.
- *[GRE]* Search the Web to evaluate your chances of getting into one of the top 25 computer science PhD programs with a GRE score of 525 Verbal and 650 Math.
- *[Computer Virus]* Search for descriptions of the next big computer virus or worm.
- *[MS Word]* In MS Word 2008 for Mac, you created a document and set the background to a grid pattern and saved it. When you opened the document later, the background no longer had the grid pattern, but was a solid color. Search the Web to find out how to resolve this.
- *[Hangar Menu]* Find the menu for the Hangar Pub and Grill in Amherst, MA.

All tasks are meant to be realistic, but are not taken from pre-existing query logs. We chose tasks we anticipated would cause some amount of frustration since our main objective is to understand frustration. Users were asked to spend no more than about ten minutes on a given task, though this was a soft deadline. A timer and reminder pop-up at the ten minute mark were provided in a browser plugin to remind them of the time.

Five of the tasks are informational, four of which are more research oriented and open ended and one that we categorize as a *technical debugging* task. The five research oriented tasks were chosen because of the anticipated time-to-completion; such open ended tasks should involve more queries and more opportunities for the user to become frustrated. The Thailand and Computer Virus tasks are the most open ended, while the Anthropology and GRE add in some additional constraints that could make the search process more difficult.

The task MS Word involves searching for the solution to a bug with Microsoft Word 2008 for Mac. The information need is informational, but what constitutes the task being satisfied depends on whether or not a proposed solution actually remedies the bug. We anticipated that formulating queries for this task would be difficult, making the user frustrated. The actual problem is real and was encountered by one of the authors.

The sixth task, Hangar Menu, is navigational. However, the source is difficult to find for those trying to find it for the first time, making this a good frustration-causing task. The inspiration for this task came from looking at query logs that contained many sessions in which users were clearly trying to navigate to a homepage or Web source that either did not exist or was unavailable.

## 4.2 Feedback

For each page that was visited for a task, the user was prompted to enter the degree to which the page satisfied the task, with an option that the page was not viewable or not in English. Five satisfaction options were given from “This result in no way satisfied the current task” to “This result completely satisfied the current task”.

After the results for a query were viewed, users were asked to assess the degree to which the results as a whole for that query satisfied their information need. We asked this because each individual result viewed may have only partially satisfied the information need, but taken together, they fully satisfy the information need. Users were also asked to assess how the results returned for the query met their expectations for the query. Five options were given, ranging from much worse to much better. Finally, users were asked to rate their frustration with the search up to the current point on a scale of 1 (not frustrated at all) to 5 (extremely frustrated).

At the end of each task, users were asked to indicate the degree to which the task was satisfied over the course of the entire session. They were also given an opportunity to comment about their knowledge of the task before they began searching.

## 4.3 Sensors

We used the mental state camera, pressure sensitive mouse, pressure sensitive chair, and features used by Cooper et al. [2] (see Section 2.2). The camera reports confidence values for 6 emotions (agreeing, disagreeing, concentrating, thinking, interested, and unsure) in addition to several raw features, such as head tilt and eyebrow movement.

The mouse has six pressure sensors that report the amount of pressure exerted on the top, left, and right sides of the mouse. Cooper et al. averaged the pressure across all six sensors to obtain one pressure reading.

The chair also has six pressure sensors: three on the seat and three on the back. Cooper et al. derived the features *netSeatChange*, *netBackChange* and *sitForward* from the raw readings.

## 4.4 Browser logging

To log both the feedback and generate a query log for the sessions, we created a Firefox plugin based on the Lemur Toolbar[1]. The events logged include the amount of a page scrolled; new tabs being opened and closed; left, right, and middle clicking on links; new windows being opened and closed; the HTML for result pages returned by Google, Yahoo!, Bing, and Ask.com; and the current page in focus.

This is a client-side query log and is richer than a server-side query log. Both client-side and server-side features can be extracted. We plan to extract features very similar to those used by Fox et al. [3].

# 5. DISCUSSION

Our initial analysis of the data from this first experiment have provided several interesting insights into modeling searcher frustration. However, we require additional experiments to provide the data necessary to make our findings statistically significant.

Across the fifteen users, a total of 351 queries were entered—an average of 3.9 per session. Users reported being frustrated (3–5 on the frustration scale) for 127 or 36% of the queries. The majority of queries (56%) performed worse than expected. Despite unmet expectations, users found their information need at least partially satisfied for 71% of queries. A total of 705 pages were visited (either from the results page or from browsing) for an average of two pages per query. Users at least partially satisfied their information need for 92% of the 90 sessions.

Figure 1 shows the level of frustration for each individual averaged over the six tasks. The x-axis shows the number of queries that have been entered so far in a session and the y-axis shows the level of frustration on the 1–5 scale. The exact frustration value is smoothed with the user’s overall average frustration, since the number of queries entered for each task is different. The thick line in the middle shows the overall average across all users and tasks. The key observations are that different users are more likely to be frustrated (or not) and frustration tends to increase as session length increases.

Looking at averages over individual tasks, there appears to be some interaction between query level satisfaction, expectation, and frustration. Tasks where users’ expectations were closer to being met/exceeded *and* queries at least partially satisfied the task also had lower frustration ratings.

Turning to individual tasks, the research oriented informational tasks shared similar characteristics in the number of queries entered, pages visited, etc. The MS Word task is an outlier in terms of information tasks, however. The task was the most frustration-invoking task, with an average frustration rating of 3.0 (moderate frustration). It also had the lowest average satisfaction rating (1.5) and meeting of expectations rating (1.7). It had the most number of queries and the second lowest number of page visitations (the first was the navigational task, Hangar Menu). The average time spent to complete or quit the task was in range of most the other tasks—about ten minutes. The high volume of queries and the low page visitation suggests that formulating queries for the task was difficult, as expected.

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

The Hangar Menu task was not as difficult as anticipated. Users entered fewer queries, visited fewer pages, and spent less time on average for this task. The average frustration rating was low (1.8) while the average satisfaction and expectation ratings were high (2.9 and 2.6, respectively).

Another interesting observation we have made is a model of frustration transitions. Aggregating across all users and tasks, we find the probability of becoming (not) frustrated given that a user is (not) frustrated. Again, we consider a user frustrated if their rating is between 3–5. This model shows that the user is frustrated after the first query in 26% of the 90 sessions and is not frustrated in the remaining 74%. Once frustrated, 82% of the time users will be frustrated after their next query and will become not frustrated 18% of the time. Once not frustrated, users will stay not frustrated after the next query 85% of the time and become frustrated the other 15% of the time. This trend is mostly consistent across individual users and tasks.

The frustration transition model gives us a key insight into understanding frustration and how to detect it. Namely, frustration is a function not only of the current interaction, but of the previous state of frustration. This suggests that a temporal classifier, such as a Hidden Markov Model, may be a good candidate for detecting frustration.

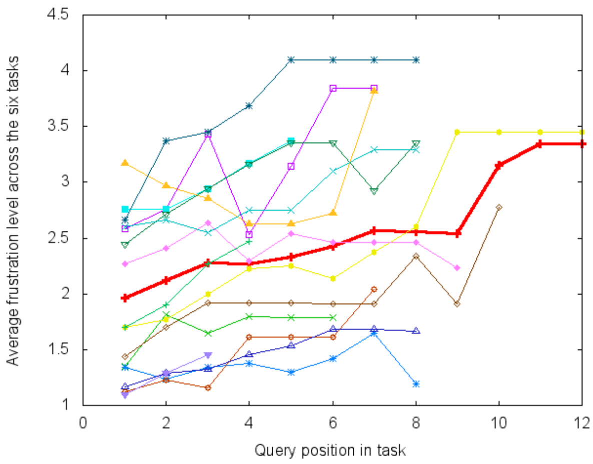


**Figure 1: The average frustration across tasks for each user after the $n^{th}$ query. Each line represents an individual user; the thick line is the average across all users.**

## 6. FUTURE WORK AND CONCLUSIONS

There are many avenues of analysis we are looking into currently, including extracting features from the query logs and sensor readings to predict frustration. Our goal is to predict frustration based solely on query logs, i.e., not to rely on sensors. We are exploring gene analysis, a technique reported by Fox et al. [3]. This form of browsing analysis abstracts the query log events, assigning a letter or symbol to a few key events. Stringing events together yields a sequence, which we can analyze in a manner similar to genes. In a brief analysis, we found that gene sequences mean different things for different tasks. For instance, the sequence "qL", meaning the user entered a query, looked at the results page and did nothing else for that query, is the most frequent sequence for two tasks and leads to frustration about 60% of the time. This probably indicates a query formulation problem. For the other four tasks, the sequence is the second or third most common sequence, but usually ends in the user not being frustrated. The same sequence leads to no or low satisfaction almost 100% of the time for the first two tasks, demonstrating a complex relationship between frustration and satisfaction.

In addition to further analysis with the current data set, we are planning a second experiment. This experiment will involve more people and different tasks. The new tasks will have a larger coverage of the informational and navigational information need types (we will not consider transactional). They will also be narrower in scope and more clearly defined. Some users in the previous experiment found the tasks too open ended. Several bugs in the logging software must also be fixed. After browser crashes, some of the JavaScript was not re-enabled, causes certain events to not be logged.

The data from our first experiment is rich and has provided us with many key insights into understanding searcher frustration. Among our observations are: frustration tends to increase with the number of queries submitted for a single task; certain searchers are more predisposed to be frustrated with the search process; a user's state of frustration is largely conserved, with a small chance of transitioning to the opposite state; and frustration appears to have a different shape for different types of tasks, such as informational versus navigational. More analysis and data will help us to understand the causes of frustration and how to model them.

## 7. ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval and in part by UpToDate. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor. We would also like to thank Beverly Woolf, David Cooper, and Winslow Burleson for loaning the sensors and logging software.

## 8. REFERENCES

[1] I. Ceaparu, J. Lazar, K. Bessiere, J. Robinson, and B. Shneiderman. Determining Causes and Severity of End-User Frustration. *International Journal of Human-Computer Interaction*, 17(3):333–356, 2004.

[2] D. G. Cooper, I. Arroyo, B. P. Woolf, K. Muldner, W. Burleson, and R. Christopherson. Sensor model student self concept in the classroom. In *First and Seventeenth International Conference on User Modeling, Adaption, and Personalization*, Trento, Italy, June 2009.

[3] S. Fox, K. Karnawat, M. Mydland, S. Dumais, and T. White. Evaluating implicit measures to improve web search. *ACM Transactions on Information Systems (TOIS)*, 23(2):147–168, 2005.

[4] A. Kapoor, W. Burleson, and R. Picard. Automatic prediction of frustration. *International Journal of Human-Computer Studies*, 65(8):724–736, 2007.

[5] R. White, J. Jose, and I. Ruthven. An implicit feedback approach for interactive information retrieval. *Information Processing and Management*, 42(1):166–190, 2006.

# Query Suggestions as Idea Tactics for Information Search

Diane Kelly
School of Information and Library Science
University of North Carolina at Chapel Hill
Chapel Hill, NC 27599-3360 USA
dianek@email.unc.edu

## ABSTRACT

This paper explores the thesis that query suggestions function as a type of idea tactic. A further thesis of this paper is that query suggestions are most useful for open-ended search tasks that require the searcher to explore and learn about a particular topic, and in situations where topics are difficult. Results are presented from two studies that examined people's use of query suggestions while searching for open-ended search topics and how usage varied according to topic difficulty.

## 1. INTRODUCTION

In 1979, Marcia Bates introduced the notion of an *idea tactic* as a move to help searchers generate new ideas or solutions to information search problems [1]. Bates noted that idea tactics serve a psychological purpose in that they are intended to help improve the searcher's thinking and creative processes (p. 280). Bates further justifies the importance of idea tactics by observing that new ideas are often "blocked or limited by one's current thinking" (p. 281). The basic idea is that the searcher's internal model of the information problem can sometimes block their efforts to think of novel and useful ways to proceed with search.

Bates proposed a number of tactics emphasizing idea generation and pattern-breaking. Idea generation focuses on the stimulation of new ideas by thinking and conducting activities outside of a retrieval system. Pattern-breaking tactics help searchers go beyond their current way of thinking about the problem and suggest moves that can be made while interacting with a retrieval system. Although some pattern-breaking tactics are intended to be used by the searcher introspectively, a number of these focus on search behavior and query generation. However, searchers' abilities to use these tactics may be limited since often searchers do not have clear understandings of their information needs [2].

Bates' article was primarily aimed at professional searchers, offering formal guidance and instruction about how to conduct bibliographic searches. Bates did not suggest how information retrieval (IR) systems might support idea tactics and since that time, few (if any) researchers have explored if and how search interfaces can support idea tactics. This paper explores the thesis that query suggestions function as a type of idea tactic by providing support for both idea generation and pattern-breaking. Query suggestions are alternative queries that the system displays to searchers. These suggestions are often identified by examining past searchers' queries and comparing these to the current searcher's query and are thus, human-generated, but suggestions can also be machine-generated. Many researchers have studied query suggestion features for information seeking tasks [e.g., 3, 6], but studies have not been conducted to understand how these suggestions support idea generation. Query suggestions can be particularly beneficial because they provide searchers with alternative methods for exploring topics and can potentially help searchers develop better understandings of their topics and richer vocabularies with which to pose queries. Furthermore, query suggestions allow searchers to continue to execute searches even when they are unable to formulate their own queries.

It is unlikely that query suggestions are useful in all types of search situations and for all types of tasks. A further thesis of this paper is that query suggestions are most useful for open-ended search tasks that require the searcher to explore and learn about a particular topic. This thesis is motivated by the information search models of Kuhlthau [4] and Vakkari [5] that depict the processes that occur while searchers engage in these types of tasks. These models are anchored by different stages, which are associated with different types and sources of desired information, search tactics and mental models. Particularly relevant to this work is Vakkari's three stages (pre-focus, formulation and post-focus) and the associated types of information sought (general information, faceted background information, specific information) and mental representations of searchers (general or vague, differentiated, integrated). It is proposed that query suggestions can assist in all stages of search by helping the searcher get started during pre-focus, explore various facets during formulation and follow-up with specific questions during post-focus. Following these models, it is further proposed that query suggestions will be most useful as an idea tactic for topics that are difficult to search.

In this paper, results are presented from two separate studies that examined people's use of query suggestions while searching for open-ended search topics. The first of these studies was presented earlier this year at the *ACM SIGIR Conference* [3], although the results presented here were not published in that paper. The second study has not been published, but is under review. In this paper, key results from each study related to the use of query suggestions as idea tactics, and compares and discusses these results. Details of the study method will not be presented, but the basic setup involved a test collection of newspaper articles (over 1 million) and assigned search tasks that asked subjects to find and save documents related to assigned topics. Subjects used an experimental search system that used Lemur[1] for indexing and retrieval. The interfaces were similar and presented subjects with a query box, basic navigation facilities and query suggestions.

## 2. STUDY 1

The major purpose of Study 1 was to evaluate the effectiveness of an experimental technique for generating query suggestions automatically that combined users' queries with terms generated by classic term relevance feedback techniques. We also compared differences between term and query suggestion interfaces. Each subject completed four topics: two with the term suggestion interface and two with the query suggestion interface. We only report results of subjects' searches with the query

[1] http://www.lemurproject.org

suggestion interface (2 searches per subject). Fifty-five undergraduate students participated in the study. Approximately half of the subjects were presented with query suggestions that had been generated automatically, while about half were presented with query suggestions that had been generated by other subjects. More details about this study can be found in [3].

## 2.1 Use of Suggestions

Subjects submitted a total of 649 queries. Five-hundred and eight (78%) of these queries were queries they entered manually while 141 (22%) were suggestions. On average, subjects entered 5.90 queries per topic (SD=4.26), or about 4.62 (SD=3.30) manually created queries and 1.28 (SD=1.92) query suggestions. The fewest query suggestions taken for a search was 0 and the most was 12. Subjects who received query suggestions that had been created by other people issued a similar number of queries to subjects who received automatically generated suggestions: 323 and 326, respectively. However, those who received user generated query suggestions selected more query suggestions (n=92) than those who received automatically generated suggestions (n=49), and entered fewer queries manually (231 vs. 277, respectively).

## 2.2 Suggestions and Topic Difficulty

The assigned search topics were classified according to difficulty, based on subjects' performances in a previous study. Topics were divided into quartiles: easy, medium, moderate and hard. Each subject completed one topic from each difficulty bin.

Table 1 displays the mean number of queries issued for topics of each level of difficulty, as well as the mean number of queries subjects generated themselves and the mean number of query suggestions taken. Overall, subjects entered more queries as topic difficulty increased. An average of 4.89 queries were entered for easy topics, while 7.50 were entered for the most difficult topics. This trend is also evident in the number of subject generated queries and the number of query suggestions, although some slight differences exist (for instance, more query suggestions were taken for moderate topics than hard topics). However, for easy and medium topics, subjects selected less than 1 suggestion per topic, while for moderate and hard topics, subjects selected about 1.65 suggestions. These results indicate that query suggestion features are more likely to be useful for difficult topics.

**Table 1. Mean (SD) number of queries entered for topics of various difficulty levels: easy, medium, moderate, and hard.**

| | Topic Difficulty | | | | |
|---|---|---|---|---|---|
| | Easy | Medium | Moderate | Hard | Total |
| Total Queries Issued | 4.89<br>(3.52) | 5.00<br>(2.77) | 6.37<br>(4.36) | 7.50<br>(5.69) | 5.90<br>(4.26) |
| Subject Generated Queries | 4.04<br>(2.93) | 4.00<br>(2.52) | 4.63<br>(2.95) | 5.92<br>(4.42) | 4.62<br>(3.30) |
| Query Suggestions Taken | 0.86<br>(1.30) | 1.00<br>(1.28) | 1.74<br>(2.64) | 1.58<br>(2.14) | 1.28<br>(1.92) |

# 3. STUDY 2

The major purpose of this study was to investigate the extent to which users could be induced to take query suggestions by manipulating usage information associated with each suggestion. The set-up of this study was similar to that of Study 1 except that subjects were only provided with user generated query suggestions. We preselected these suggestions from queries entered by subjects in Study 1 for the same search topics (only four topics were used in Study 2). Eight query suggestions were provided for each topic – four query suggestions were good performing queries and four were poor performing queries (we predetermined good and poor queries by examining how many relevant documents were retrieved in the top 20 results). Usage information indicating how many other people used the queries was also displayed next to the query suggestions. For half of the queries this information was high (i.e., many people used the query) and for half the queries this information was low (i.e., very few people used the query). The order of the queries and associated usage information was randomized. Twenty-three subjects participated (22 undergraduates and 1 graduate).

## 3.1 Use of Suggestions

Subjects submitted a total of 722 queries for all topics combined (32 queries on average, or about 8 queries per subject per topic). Four-hundred twenty-five (59%) of these queries were of their own creation while 297 (41%) were suggestions. Each subject was shown a total of 32 suggestions (8 suggestions per topic * 4 topics) and selected an average of 13.70 (*SD*=7.02). One subject did not select any suggested queries, while another selected 24[2].

The number and proportion of query suggestions taken by subjects in this study (n=297, 41%) was much greater than the number taken by subjects in Study 1 (n=141, 22%). Although many aspects of these studies were similar (subject population, system, collection, topics, basic interface), several main aspects (such as the content of the query suggestions) differed. Thus, we are unable to make any conclusive statements about what impacted the difference in use of suggestions, but note one major difference – subjects' perceptions of the origins of the suggestions. In Study 1, some query suggestions were created by other users and some were created automatically by the system, but subjects were not provided with any information about the origins of these suggestions. In Study 2, subjects were only presented with user generated suggestions and were told via the interface the origins of the suggestions (that is, subjects were told that certain numbers of other users entered the queries). We did not quiz subjects in the first study to find out their beliefs about the origins of the query suggestions, so we cannot say with certainty that they did or did not believe the queries were created by humans. However, the discrepancy in the amount of suggestions taken between subjects in the two studies suggests that agency attribution is an important factor in determining whether subjects take suggestions and that attributing the suggestions to other humans rather than a machine might increase the use and uptake of suggestions. Our future work will explore this directly by manipulating the actual and communicated origins of the query suggestions. Results of such work might demonstrate if users are biased towards human recommendations and why they appear to make more use of these recommendations than computer generated recommendations.

Since the set of suggested queries for each search remained the same, we were able to examine the overlap between the queries

[2] Subjects in this study were not influenced by the usage information: subjects selected 148 queries that were associated with low usage information and 165 queries that were associated with high usage information.

subjects entered manually and the suggested queries to see how many duplicates occurred. That is, how many times a manually entered query matched exactly a suggested query. We found that 113 of the 425 subject-created queries were exactly the same as one of the suggested queries. Of these 113 queries, only 7 were the first queries typed by subjects (in which case they would not be duplicates since subjects did not receive any suggestions until they entered one query). Although there is no way to ascertain that subjects did not naturally type the 106 duplicate queries on their own, these results suggest that some subjects took suggestions by manually entering the queries rather than clicking on them. Thus, using a simple click metric to measure uptake may not tell the whole story. While suggested queries may provide the subject with ideas for search and a greater understanding of their topic, this usage may not easily measurable by observing behavior. Eye tracking can potentially provide a better indicator of how frequently a subject examines suggestions and can also be correlated with observable behaviors, but this too does not adequately capture the extent to which query suggestions expand the user's knowledge of the topic and potentially assist with query formulation and identifying search moves. Thus, one challenge to studying the use of query suggestions as idea tactics is identifying more robust methods for observing their impact.

## 3.2 Integration of Suggestions into Searching

When do subjects use query suggestions during their searches? Figure 2 shows the sequence of queries entered by subjects and whether the queries were created by the subject or were suggestions. (Note that subjects had to enter one self-created query at the start of the session before any suggestions were displayed.) This Figure shows that many subjects made use of the suggestion feature early in their searches. For the second and third queries issued, subjects were nearly equally as likely to enter their own query or click on a suggestion.

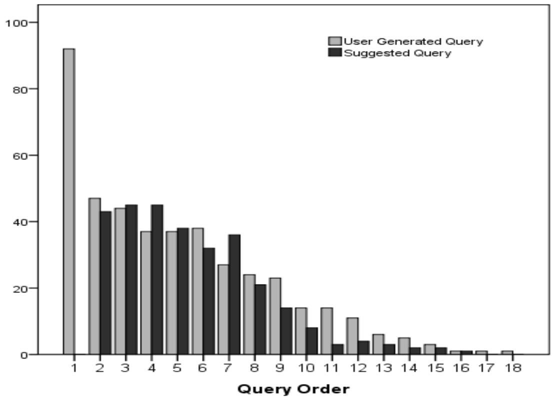


**Figure 2. Frequency and source of query (self-created or suggested) according to order of submission during search.**

One remarkable thing about this figure is the number of queries that were submitted: in one case a subject entered 18 queries during a search, which is much more than what is commonly reported in the literature. In Study 1, the maximum number of queries issued by a subject during a search was 22, an even greater number. On average, subjects in Study 1 issued about 5.90 queries per topic, while those in Study 2 issued about 8 per topic. While there are many possible explanations for subjects entering larger numbers of queries than usual, including the fact that they were in a laboratory study, it may be the case that the query suggestions caused the increase. The suggestion may have supported subjects' query behavior explicitly by providing them with clickable suggestions, or implicitly by providing them with ideas which they could use to create their own new queries.

In Figure 3, manually entered queries that duplicated a query suggestion are counted as a user generated query. However, if we consider these as suggestions (Figure 2), we see that subjects integrated the suggestions into their searches even more quickly – about 65% of the second queries issued by subjects were suggestions. For the second to eighth queries entered, subjects were more likely to use a suggestion than enter their own queries, while those who continued past the eighth query were more likely to create their own queries in subsequent iterations.

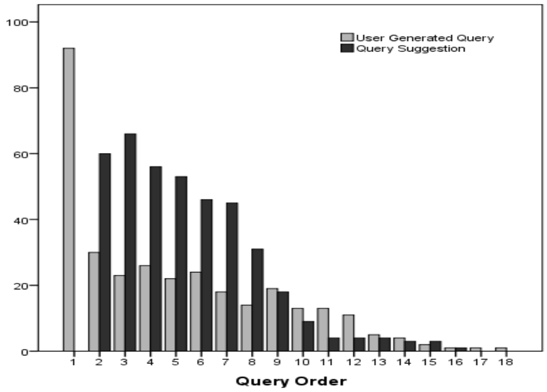


**Figure 3. Frequency and source of query (self-created or suggested) according to order of submission during search. In this Figure, user generated queries that duplicated query suggestions are counted as query suggestions.**

## 3.3 Suggestions and Search Stage

To understand more about when subjects were more likely to take suggestions, we divided subjects' searches into three equal parts: beginning, middle and end. This division was based on the length of time each individual subject searched for a specific topic. For example, if one subject spent 12 minutes searching for one topic, then the beginning stage corresponded to the first 4 minutes of the search, the middle stage the next four minutes and the end stage the last four minutes. Of course, such divisions do not correspond temporally to the stages identified by Vakkari [5] who studied searchers for several months. However, we assume that even within a short 15 minute search that subjects will execute similar 'mini' stages of pre-focus, formulation and post-focus where they are more likely to be looking for general information in the pre-focus stage, faceted background information in the formulation stage and specific information in the post-focus stage.

Figure 4 displays the number of user generated queries issued and suggested queries taken for each search stage. In the first Stage, subjects entered more of their own queries, but during Stages 2 and 3, subjects selected more suggested queries. This difference is especially evident at Stage 3. These results suggest that query suggestions might be particular useful during the latter stages of search when subjects are spending more time exploring the various facets of the topic and following-up by looking for specific information. These are likely to be the points at which subjects have exhausted their own ideas for queries and need ideas for alternative queries to continue with their searches.

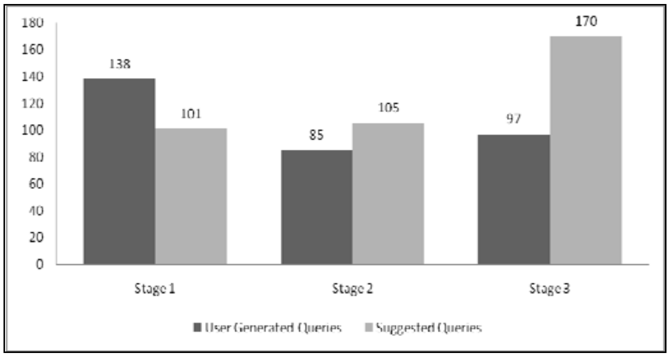


**Figure 4. Number of user generated queries issued and suggested queries taken for each search stage.**

## 3.4 Suggestions and Topic Difficulty

We were also interested in seeing if we could replicate the finding from Study 1 with regard to use of suggestions and topic difficulty. Recall that in Study 1, subjects issued more queries and took more suggestions for more difficult topics. Figure 5 displays the number of user generated queries issued and the number of suggested queries taken according to topic difficulty. These difficulty rankings are based on subjects' post-search evaluations of how difficult it was to search for the topic.

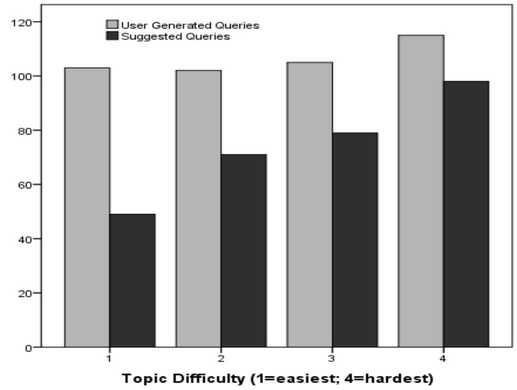


**Figure 5. Relationship between topic difficulty and number of user generated queries and suggested queries.**

The relationship between search difficulty and use of suggestions was even more pronounced than in Study 1, with subjects taking twice as many suggestions for the most difficult topic than the easiest topic. Similar to Study 1, subjects issued more queries for more difficult topics and the number of queries subjects created on their own was somewhat constant. However, the proportion of query suggestions taken to total queries issued saw greater increases in Study 2 as a result of topic difficulty than in Study 1.

The results of both studies suggest that query suggestions might be useful for difficult topics since subjects in both studies issued more queries for difficult topics. For easy topics, subjects may find a sufficient number of documents by issuing a small number of their own queries or may be able to think of a variety of queries based either on their pre-search understanding of the topic or on their search interactions. For difficult topics, subjects may need more assistance both at the beginning and latter search stages because of a lack of pre-search topic knowledge or because initial queries do not yield useful results. One factor that may contribute to search difficulty is the complexity of the topic, where complexity is the number of topic facets. If searchers are unaware of the range and types of facets, then they are likely to have a difficult time identifying queries that represent these different facets. Query suggestions can potentially provide users with an overview of different facets as well as easy entry points into different parts of the collection. Facet and cluster-based browsing is not new, but browsing via query suggestion may provide a more meaningful and effective method of access for users.

## 4. CONCLUSION

The thesis that query suggestions function as a type of idea tactic was explored using data from two studies. Results showed that subjects used more query suggestions when searching for difficult topics and during the latter stages of search. Results also showed that subjects integrated suggestions into their searches fairly quickly and that they often manually enter suggested queries. These results provide preliminary support for the notion of query suggestions as idea tactics, although further study is necessary.

Bates [1] states that the ultimate measure of success for an idea tactic is whether it improves retrieval performance. Our next step is to investigate search performance. This includes investigating overall performance, individual query performance and cumulative, within-session performance. A direct comparison of searchers' interactions and tactics with a system that provides query suggestions with one that does not would also help us better understand the potential usefulness of suggestions as idea tactics. Such a study would allow one to explore if searchers using a query suggestion system searched longer, issued more queries (and at different rates, at different stages), achieved better performance, and learned more about their topics than searchers using a system that did not offer suggestions.

Additional studies should examine different methods for exploring the usefulness of query suggestions as idea tactics since performance is only one outcome. As Bates [1] described, idea tactics primarily serve a psychological purpose. A better understanding of how to evaluate query suggestions with this goal in mind is an important area for future research.

## 5. ACKNOWLEDGEMENT

Many wonderful students helped with the studies: Karl Gyllstrom, Earl W. Bailey, Xi Niu, Amber Cushing and Maureen Dostert.

## 6. REFERENCES

[1] Bates, M. J. (1979). Idea tactics. *Journal of the American Society for Information Science, 30*, 280-289.

[2] Belkin, N. J. (2000). Helping people find what they don't know. *Communications of the ACM, 43*(8), 58-61.

[3] Kelly, D., Gyllstrom, K., & Bailey, E. W. (2009). A comparison of query and term suggestion features for interactive searching. *Proceedings of SIGIR '09*, 371-378.

[4] Kuhlthau, C. C. (1993). *Seeking meaning: A process approach to library and information services*. NJ: Ablex.

[5] Vakkari, P. (2004). Changes in search tactics and relevance judgments when preparing a research proposal: A summary of the findings of a longitudinal study. *Information Retrieval*, *4*(3-4), 295-310.

[6] White, R. W., Bilenko, M., & Cucerzan, S. (2007). Studying the use of popular destinations to enhance web search interaction. *Proceedings of SIGIR '07*, 159-166.

# I Come Not to Bury Cranfield, but to Praise It

Ellen M. Voorhees
National Institute of Standards and Technology
Gaithersburg, MD USA
ellen.voorhees@nist.gov

## 1. INTRODUCTION

Much information retrieval research is currently performed using test collections, a methodology introduced by Cleverdon and his colleagues in the Cranfield tests [4] and further refined in evaluation exercises such as TREC (http://trec.nist.gov/). A test collection is (purposely) a stark abstraction of real user search tasks that models only a few of the variables that affect search behavior and was explicitly designed to minimize individual searcher effects. Nonetheless, I argue that Cranfield-style experimentation is critical to the study of interactive (user-in-the-loop) retrieval for at least two reasons. First, research using test collections identifies good retrieval technology, allowing expensive user testing to be reserved for the most promising avenues. Second, meta-analysis of the Cranfield methodology can inform the development of new research abstractions that make different trade-offs among realism, experimental power, and cost.

This paper is a condensation of an earlier paper in which I made similar arguments [12]. The arguments stem from my experience with building and validating test collections as the manager of the TREC project, a role that clearly marks me as a Cranfield advocate. The next section provides a brief recap of the Cranfield methodology as currently practiced, the following section examines the immense impact of user variability on retrieval experiments, and the final section describes two prior TREC efforts to develop evaluation paradigms in the space between test collections and full interactive experiments. The paper does not contain a proposal for a new abstraction for testing interactive retrieval systems; that is very much an open research problem. Instead, my hope is that the paper clarifies some of the issues that must be addressed by such an abstraction.

## 2. THE CRANFIELD PARADIGM

A Cranfield test collection is a set of documents, a set of information need statements (called "topics" in the remainder of this paper), and a set of relevance judgments that list which documents should be returned for which topic. In the simplest and most common case, relevance judgments are binary, either a document is relevant to the topic or it is not, and are based on topicality, a document is relevant if its subject matter matches that of the topic.

In current practice, a researcher runs a retrieval system on a test collection to produce a ranked list of documents in response to each topic (a "run"). A ranking reflects the system's idea of which documents are likely to be relevant to the topic; documents it believes more likely to be relevant are ranked ahead of documents it believes are less likely to be relevant. Using the relevance judgments, various evaluation metrics can be computed for the ranked list, each of which reflects some aspect of the goal of ranking all relevant documents ahead of all nonrelevant documents. Scores for individual topics are averaged over the set of topics in the test collection. Average scores for one run are compared to other runs using the exact same test collection. Retrieval systems producing runs with better scores are considered more effective retrieval systems.

The abstraction that a test collection implements is admittedly impoverished compared to the complexity of a real search task. This is a deliberate design choice to provide more control over experimental variables at the cost of less realism. Performing well on this abstract task is assumed to be a necessary *but not sufficient* prerequisite for performing well on real search tasks. If a retrieval method can't at least rank relevant documents before nonrelevant documents (for some reasonable definition of relevance), it is hard to imagine how it can be successful at any real user task. Test collections are thus convenient tools for studying retrieval system performance in the laboratory. As in medical research, laboratory research plays an integral role in the development of new treatments, weeding out less successful approaches early and relatively inexpensively, while reserving much more costly ("clinical") testing of more realistic functionality for only the most promising approaches.

Of course laboratory research is only useful if its findings generalize to real-world tasks. There is plenty of historical evidence that Cranfield-style tests are useful. Basic components of current commercial retrieval systems, including full text indexing, term weighting schemes, and relevance feedback, were first developed using test collections. And while Hersh and Turpin conclude that relevance ranking in the manner of Cranfield is not a trustworthy means to find differences in systems that matter to users [5, 11], I contend their studies are better interpreted as examples of the difficulty of gaining sufficient power to distinguish among systems in interactive experiments.

*HCIR'09,* October 23 2009
ACM X-XXXXX-XX-X/XX/XX.

What is the source of this lack of power? Variability. Variability and experimental power are inversely related in any experimental design. An experimental design that provides a given level of power will be increasingly expensive as variability increases since more factors must be controlled for. As described in the next section, users are the dominant source of variability in retrieval experiments *even in the Cranfield paradigm.* Any additional user attributes included in an experimental design can only increase that variability.

## 3. USER EFFECT AND VARIABILITY

The user is represented in a test collection simply as the combination of a topic and its corresponding set of relevance judgments. This ruthless abstraction of the user to a static topic statement is a leading cause of Cranfield critics' frustration with the methodology. Yet the topic (i.e., the information being sought) is the primary source of variation in Cranfield-style retrieval experiments. An analysis of variance model fitted to the TREC-3 results demonstrated that the topic and system effects, as well as the interaction between topic and system, were all highly significant, with the topic effect as the largest [2]. In other words, even when the user is reduced to simply the question asked, differences between users have a bigger effect on the outcome of an experiment than do differences between systems.

The large variability in topic performance for a single retrieval system is precisely why the Cranfield paradigm uses average scores over a set of topics in comparisons. Seasoned experimenters have long known that the topic set needs to be relatively large. For example, in the mid-1970's Sparck Jones and van Rijsbergen suggested 75 topics as a minimum number of topics for an 'ideal' test collection[9]. This intuition arose from observing the behavior of individual topics as demonstrated in Figure 1. The figure shows an interpolated precision-recall graph for an example TREC run. The heavy solid line is the average recall-precision curve over the 50 topics in the test set, while the dotted lines are the curves for 15 individual topics within the test set.

The archives of retrieval results from programs like TREC and NTCIR allow hypotheses regarding test collection construction and use (such as minimum topic set size) to be experimentally verified. Investigations have established an empirical relationship between the number of topics in a test set, the size of the difference in retrieval scores used to decide two runs are different ($\delta$), and the error rate [13, 8], as well as demonstrated that some evaluation measures are more stable than others (including that MAP is much more stable than precision at a given cut-off) [3]. For all measures looked at, the error rate decreases when either the topic set size or $\delta$ increases. For the most stable measures and for $\delta$'s of the size commonly observed in the retrieval literature, 25 topics is clearly too few to have confidence in the conclusion, and even 50 topics is somewhat iffy—a $\delta$ of between 0.03 and 0.04 in MAP scores, equivalent to about a 10% relative difference in MAP scores for current systems, had an error rate of about 11%. Less stable measures have higher error rates for equal topic set sizes.

These are sobering conclusions for the prospects of building user-focused abstractions that provide reliable comparisons among systems. Even in the most controlled environment—the Cranfield task and using MAP as the measure—you still need on the order of 50 topics to control user variability sufficiently to have confidence in a system comparison. Modifications as small as moving from MAP to a more user-focused measure like precision at ten documents retrieved require larger topic sets for a similar level of confidence. More radical departures will require even larger topic sets. Robertson calculated that hundreds of topics per user would be needed to obtain statistical significance in non-matched-pair tests[7].

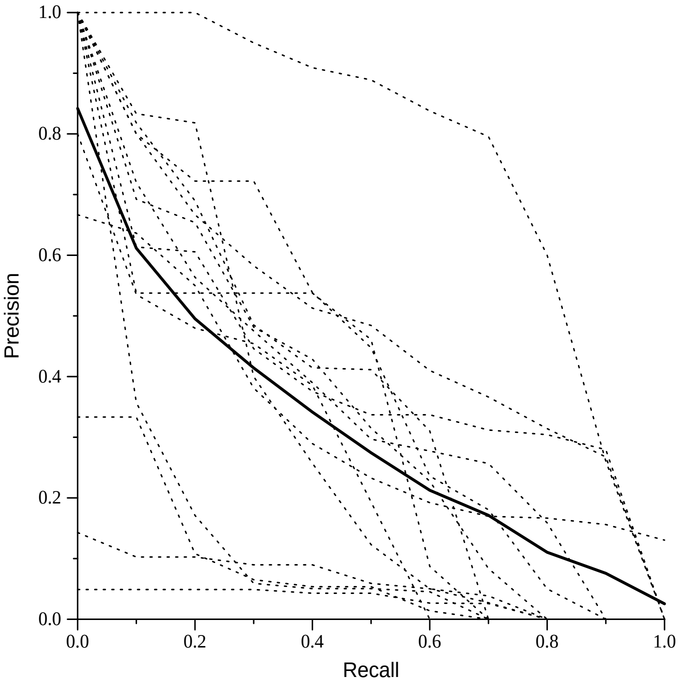


**Figure 1: Interpolated precision at standard recall points for selected individual topics (dotted lines) and the overall average for the run (solid line).**

## 4. CRANFIELD ALTERNATIVES

So far I have argued that the Cranfield paradigm is successful because of its carefully calibrated level of abstraction. The document ranking task has sufficient fidelity to real user tasks to be informative, but is sufficiently abstract to be broadly applicable, feasible to implement, and comparatively inexpensive.

Interactive studies, currently the main alternative to test collection experiments, generally have substantially more fidelity to real user tasks than Cranfield tests. Unfortunately, interactive studies that are powerful enough to reach meaningful conclusions are also challenging to design and expensive to implement. The difficulties arise for a variety of reasons. Interactive experiments have a large start-up cost. Regardless of what particular functionality is the actual object of interest, a good interactive experiment requires complete systems that support the functionality in its best light for all alternatives. Large numbers of topics are required to reach statistically significant conclusions. Large numbers of topics imply large numbers of subjects and sophisticated experimental designs to balance subjects across conditions (any given subject can search a given topic no more than once). The more specific the design is to a given operational setting (i.e., the more "real" the experiment) the less generalizable the findings are to other environments so the more cases that need to be examined. Tague lists a myriad of factors that must be considered when designing retrieval experiments [10].

The quest, then, is to find useful abstractions that make different trade-offs among realism, experimental power, and cost than test collections on the one hand and interactive studies on the other. TREC has contained two notable efforts in this regard. The TREC-6 interactive track looked to decrease the cost of cross-site interactive tests by using a common baseline system, while the TREC HARD track augmented a traditional test collection with additional user information.

In the TREC-6 interactive track [6], each participating site ran an experiment comparing their system to a common baseline system. The assumption in the design was that this set-up would allow the effect of performing all $n^2$ comparisons among the $n$ participating sites for a total cost of only $n$ comparisons (one at each site) since the common system would control for inter-site variance. However, subsequent analysis of the results did not support the assumption. Further, the design incurred its own costs: participating sites had to obtain, install, and support the common system; and precious human subjects had to be devoted to the common system, a system that was incidental to the main purpose of the experiment being run at that site. As a result of these limitations, the design could not be recommended.

The overall goal of the TREC HARD ("High Accuracy Retrieval from Documents") track was to improve ad hoc retrieval by customizing the search to the particular user. The motivation for the track was that current retrieval systems return results for the "average" user and this necessarily limits their effectiveness for the particular user. The task in the track was an ad hoc document ranking task but with extra information available at search time.

The TREC 2004 instantiation of the track [1] provided two different sources of additional information, metadata supplied in the topic statement and information collected from so-called clarification forms. There were five categories of metadata: the (putative) user's familiarity with the subject area specified as either *little* or *much*; the genre of the documents sought specified as either *news-report*, *opinion-editorial*, *other*, or *any*; the geographical location of the source of the document (*US*, *non-US*, or *any*); a short free-text description of the subject domain; and text from related documents if available. Clarification forms were HTML-based forms that contained a task for the user (in this case, the TREC assessor) to perform. Track participants were free to implement in the form any task they thought would help retrieval, subject to the constraints that the assessor would spend no more than three minutes per topic responding to a form and the form had to be completely self-contained. Examples of clarifying forms were asking the user to resolve the senses of ambiguous topic words or obtaining relevance judgments on terms or document snippets.

The track protocol required participants to perform baseline runs using only the traditional topic statement. If desired, they could then submit clarification forms to receive the results of the assessor using the forms. Finally they performed additional (non-baseline) runs using the metadata from the extended topic statements and/or the clarification form results. This protocol allowed direct comparison between standard and extended information runs for a single participant as well as comparisons among participants. Figure 2 gives a precision-recall graph of the top-performing TREC 2004 HARD track runs including one baseline and one additional run per top group. A pair of runs plotted with the same symbol are a pair submitted by a single group. Baseline runs are plotted using a dotted line, and additional runs are plotted using a solid line.

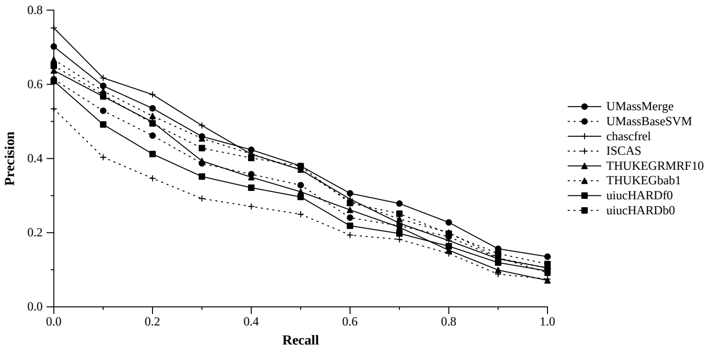


**Figure 2: Effectiveness of top scoring runs submitted to the TREC 2004 HARD track.**

In general (though not invariably) the additional information did improve retrieval effectiveness over the corresponding baseline run. But understanding what factors actually contributed to the improvement is difficult. There are too few topics in a single metadata category (for example, familiarity) to draw reliable conclusions regarding category-specific retrieval techniques. Use of clarification form data by researchers other than the original submitters is complicated by needing to understand the specifics of the forms and the inability to repeat the experiment with slight variations. Little reuse means the costs of creating the resource is not leveraged over the wider community. Since the construction of the HARD collection was already more expensive than standard Cranfield collections (the metadata categories needed to be decided on and then topics created that populated those categories; researchers needed to create clarification forms and assessors had to fill them out), the lack of reusability is a double penalty.

These were good attempts at defining a new evaluation paradigm for interactive retrieval, though in the end neither approach was as successful as hoped. The HARD track underscores how a fairly small change in the amount of user information incorporated into the experimental design mushrooms into a much larger set of experimental conditions. The number of topics needed to control for user variability to reach reliable conclusions is truly daunting. I believe that any new evaluation paradigm that attempts to encompass all/most/many of the factors that can affect search behavior is doomed to fail. Instead, like Cranfield, a successful new paradigm will need a carefully defined abstract task that models only a minimum of of critical features.

## 5. REFERENCES

[1] James Allan. HARD track overview in TREC 2004: High accuracy retrieval from documents. In *Proceedings of the Thirteenth Text REtrieval Conference, TREC 2004*, pages 25–35, 2005. NIST Special Publication 500-261.

[2] David Banks, Paul Over, and Nien-Fan Zhang. Blind men and elephants: Six approaches to TREC data. *Information Retrieval*, 1:7–34, 1999.

[3] Chris Buckley and Ellen M. Voorhees. Evaluating evaluation measure stability. In N. Belkin, P. Ingwersen, and M.K. Leong, editors, *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 33–40, 2000.

[4] C. W. Cleverdon, J. Mills, and E. M. Keen. Factors determining the performance of indexing systems. Two volumes, Cranfield, England, 1968.

[5] William Hersh, Andrew Turpin, Susan Price, Benjamin Chan, Dale Kraemer, Lynetta Sacherek, and Daniel Olson. Do batch and user evaluations give the same results? In N. Belkin, P. Ingwersen, and M.K. Leong, editors, *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 17–24, 2000.

[6] Eric Lagergren and Paul Over. Comparing interactive information retrieval systems across sites: The TREC-6 interactive track matrix experiment. In W. Bruce Croft, Alistair Moffat, C.J. van Rijsbergen, Ross Wilkinson, and Justin Zobel, editors, *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 164–172, Melbourne, Australia, August 1998. ACM Press, New York.

[7] S. E. Robertson. On sample sizes for non-matched-pair IR experiments. *Information Processing and Management*, 26(6):739–753, 1990.

[8] Tetsuya Sakai. Evaluating evaluation metrics based on the bootstrap. In *Proceedings of the Twenty-Ninth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2006)*, pages 525–532, 2006.

[9] K. Sparck Jones and C.J. van Rijsbergen. Information retrieval test collections. *Journal of Documentation*, 32(1):59–75, 1976.

[10] Jean Tague-Sutcliffe. The pragmatics of information retrieval experimentation, revisited. *Information Processing and Management*, 28(4):467–490, 1992.

[11] Andrew H. Turpin and William Hersh. Why batch and user evaluations do not give the same results. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 225–231, 2001.

[12] Ellen M. Voorhees. On test collections for adaptive information retrieval. *Information Processing and Management*, 44(6):1879–1885, 2008.

[13] Ellen M. Voorhees and Chris Buckley. The effect of topic set size on retrieval experiment error. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 316–323, 2002.

# Search Tasks and Their Role in Studies of Search Behaviors

Barbara M. Wildemuth
School of Information & Library Science
University of North Carolina at Chapel Hill
Chapel Hill, NC 27599-3360, USA
wildem@ils.unc.edu

Luanne Freund
School of Library, Archival & Information Studies
University of British Columbia
Vancouver, BC, V6T 1Z1, Canada
luanne.freund@ubc.ca

## ABSTRACT

In experimental studies of search behaviors and evaluations of information retrieval systems, researchers generally assign search tasks to the subjects to perform. Since it can be expected that the tasks themselves will influence search behaviors and performance, we need to be able to construct tasks having particular attributes, knowing that our study findings can then be generalized to all search tasks having those attributes. In this paper, we report on an ongoing analysis of the search tasks that have been used in experimental search studies. We review a number of typologies of search tasks currently in use (complex vs. simple, specific vs. general, exploratory vs. lookup, and navigational vs. informational) and make recommendations for designing search tasks for use in future studies.

## Categories and Subject Descriptors

H3.3. Information search and retrieval: Search process

## General Terms

Experimentation

## Keywords

Search tasks, Research design

## 1. INTRODUCTION

People's search behaviors vary widely. It's likely that some of this variation is *not* related to differences in the characteristics of individual searchers (e.g., domain knowledge or search expertise), but is instead due to differences in the goals that they are trying to achieve. In almost all cases, searches for information are undertaken within the context of some other purpose, goal, or activity. In other words, the person's search behaviors are situated within the context of performing some larger task [30, 33].

These embedding tasks may vary along a number of dimensions, including their complexity, structure, and granularity. For example, consider the difference between the complex and amorphous task of completing a dissertation and the simple and well-structured task of locating the address of a local store. While it is important to understand the characteristics of these tasks, by which people's search behaviors are motivated [8, 29], it is also important to focus on the attributes of the search tasks[1] themselves, and how those attributes can affect search behaviors.

In particular, it is important to understand the potential influence of the search tasks assigned to research subjects when studying search behaviors and evaluating information retrieval (IR) systems. Naturalistic studies are intended to observe real people searching in order to complete real tasks; however, experimental studies are intended to isolate particular effects on user behaviors. Because of this desire for control in experimental studies, researchers usually assign search tasks, either controlling the task effect by assigning the same tasks to all the subjects or manipulating it as an independent variable. Most studies to date have opted for control over manipulation, but in either case, researchers are handicapped by our lack of understanding of the influence of the search task on the study findings. Given that search tasks can vary along many dimensions, findings may be valid for a particular set of tasks, but we do not know to which additional tasks they may be validly applied.

In order to make additional progress in experimental studies, we need to gain a better understanding of search tasks and their effects.  We need to be able to construct tasks having particular attributes, knowing that our findings can then be generalized to all search tasks having those attributes. To this end, we are collecting and analyzing the search tasks that have been used in experimental search studies.[2] This paper is proposed as a starting point for gaining an understanding of these tasks. We will briefly review a selection of studies and compare the ways in which search tasks have been categorized in those studies. We will conclude with suggestions for moving forward on this research agenda.

## 2. TYPES OF SEARCH TASKS

When designing a study of search behaviors, the researcher needs to decide how much control to exert over the search tasks. At one end of the spectrum, the study subject is allowed to search on tasks of personal interest (i.e., the tasks are fully self-generated [3] or natural [30]). At the other end, the tasks are fully specified by the researcher. Some studies use a combination of tasks generated

*HCIR 2009*, October 23, 2009, Washington, DC

[1] In this paper, the term, search tasks, will be used to designate the goal(s) to be achieved in a specific search situation. They are distinguished from the more general (work-related or other) tasks that have motivated the search. This distinction is explained in more detail by Byström and Hansen (2005).

[2] So far, we have collected over 223 descriptions of tasks from 79 empirical studies and conceptual papers. In addition, we are examining conceptual papers discussing search tasks.

by the researchers and tasks generated or modified by the subjects [28]. In this discussion, we are concerned only with the tasks generated and assigned by researchers.

In some past studies, the researchers have not specified the attributes of the assigned search tasks. In other studies, the researchers have described or categorized the tasks in some way (e.g., as complex, simple, known-item, factual, exploratory, navigational, or informational [18]). Attempts to integrate these typologies include Bilal's [3] integration of open- and closed-ended tasks with complex and simple tasks; Li and Belkin's [22] faceted classification of tasks; and Jansen, Booth, and Spink's [16] investigation of Broder's [6] original typology of Web search tasks as information, navigational, or transactional. This section will discuss some of the ways in which search tasks have been categorized in recent studies of search behaviors, providing specific examples of each. This review of varying typologies of search tasks is meant to be suggestive of future research directions, rather than exhaustive.

## 2.1 Task Attributes Not Specified

In many studies, assigned search tasks are clearly described and, possibly, the full text of the search tasks is provided, yet the tasks are not categorized as being of a certain type. For example, in Woodruff et al.'s [34] comparison of three types of thumbnails during Web searches, the researchers developed and assigned search tasks that were "much like typical Web search tasks" (p.176). The 12 search tasks covered four areas: picture, homepage, e-commerce, and side effects. These groupings may be interpreted as connoting the topic of the search; for example, the side effects category included the search task, "Find at least three side effects of Halcion" (p.177).

Other authors might have categorized these search tasks differently. For example, the Halcion example might be classified as a complex search task, since it is likely that the searcher will need to consult several different Web pages to find multiple side effects. It might also be classified as a factual search task, since facts are the end point of the search process.

## 2.2 Complex vs. Simple Tasks

Complexity is the most commonly manipulated attribute of search tasks, although it has been defined and operationalized in many different ways [29]. Unlike other more discrete variables, complexity tends to be treated as an aggregate of one or more of the following task characteristics: structure [21, 26], certainty or *a priori* determinability [1, 7, 8, 10], number of facets [11, 2], length of the search path [11, 14], cognitive effort [2, 11] and topic familiarity [2, 7].

As an example of studies of task complexity, Bell and Ruthven [1] undertook a study that drew upon earlier work by Byström [8] and Campbell [10]. They developed sets of three tasks on the same topics but at differing levels of complexity by manipulating the degree of uncertainty. For the most complex task, "it is unclear what information is being sought, how to obtain relevant information, and how the searcher will know they have found relevant information" (p.61). For example, one of the low complexity task scenarios asks the searcher to "find out how the price of petrol in the UK has changed in recent years," while the corresponding high complexity task asks the searcher to "find out how and why petrol prices vary worldwide" (p.62).

The broad range of conceptualizations of complexity can be seen by comparing Bell and Ruthven's most complex task example with that of the most complex task assigned by Browne, Pitts and Wetherbe [7]. They also assigned tasks at three levels of complexity, and the most complex was to find a map of a little-known battlefield. While this task proved to be relatively difficulty to perform, by many definitions of complexity, it is simpler than the simple tasks assigned by Bell and Ruthven [1].

## 2.3 Specific vs. General Tasks

The specificity of the assigned search tasks is another task attribute that researchers have manipulated. Across studies, "specific" tasks tend to have more clearly defined goals than "general" tasks. Specific tasks may be equated with known-item search tasks (e.g., in [19]), factual tasks (e.g., in [14]), or simple lookup tasks (e.g., in [11, 13]).

Rouet [25] focused a study on the effects of task specificity on searching behaviors. The specific search tasks were defined as asking the study subjects to "locate one piece of information" (p.415); an example is, "Which authors have provided the first clinical descriptions of anorexia"?, to be searched in a hypertext document on anorexia. The general search tasks were defined as requiring "the reading and integration of 2-5 separate passages" from the hypertext document; an example is, "What treatments [for anorexia] may be suggested, and what are their effects?" While it is certainly appropriate to describe these two types of search tasks as specific and general, other researchers might have described them as simple and complex.

## 2.4 Exploratory vs. Lookup Tasks

Over the past several years, interest in exploratory search behaviors has increased. Users conducting exploratory searches are likely to "submit a tentative query and take things from there, exploring the retrieved information, selectively seeking and passively obtaining cues about where the next steps lie" [31, p.38]. Exploratory searching is defined as searching that supports learning, investigating, comparison, or discovery [20, 24]. It is contrasted with lookup tasks, which are oriented toward finding particular facts or answering specific questions. There is some evidence that this contrast is perceived by searchers as well as researchers, as found by Kules and Capra [20].

White and Marchionini [32] incorporated both exploratory and lookup tasks in their study of a new approach to query expansion. An example of a lookup task was, "You are doing some research for a term paper you are writing and need to find the name of the first woman to travel in space and her age at the time of her flight"; an example of an exploratory task was, "You are about to depart on a short-tour along the west coast of Italy. The agenda includes a visit to the country's capital, Rome, during which you hope to find time to pursue your interest in modern art. However, you have recently been told that time in the city is limited and you want information that allows you to choose a gallery to visit" (p.689). Interest in exploratory search behaviors and retrieval systems that can better support exploratory search is likely to continue. While Kules and Capra [20] have made a significant contribution in this direction, additional studies are needed to more clearly conceptualize and define exploratory search tasks.

## 2.5 Transactional vs. Navigational vs. Informational Tasks

The types of tasks already discussed are intended to represent people's information needs [15, Fig.6.8]; they all assume that a person searches because that person needs to find information for a particular purpose. Broder [6] argues that, with the advent of

Web searching, we need to broaden our perspective. In addition to informational tasks, people also may be using the Web to accomplish navigational or transactional tasks. While the purpose of an informational task is to acquire some information, the purpose of a navigational task is "to reach a particular site" and the purpose of a transactional task is "to perform some web-mediated activity" (p.5). Examining query logs from AltaVista, Broder found that about 48% of Web queries were informational, while 20% were navigational and 30% were transactional. Jansen, Booth, and Spink [16] followed up on Broder's work with an attempt to automatically classify Web queries into Broder's three types. In addition, they provided a more fine-grained analysis of Broder's three types of Web queries.

This classification of pre-existing Web queries is now beginning to serve as an empirical basis for studies in which search tasks are designed and assigned to study subjects. For example, Lorigo et al. [23] assigned five navigational tasks and five informational tasks to their subjects in a study of subjects' behaviors during searching and review of results, and Joachims et al. [17] assigned the same 10 tasks to their subjects in a study of the implicit feedback provided by user clicking behavior. An example navigational task was, "Find the homepage for graduate housing at Carnegie Mellon University"; an example informational task was, "What is the name of the researcher who discovered the first modern antibiotic?" (p.6). All of the questions are also depicted as closed-ended, and so did not cover the full range of informational search tasks described in the previous sections.

This typology was useful in developing the two example studies described. However, neither study incorporated transactional tasks. In addition, the findings cannot be generalized to all types of informational tasks because only a homogeneous set of straightforward factual search tasks were assigned.

## 2.6 What We Can Learn from the Current Task Typologies

The few typologies discussed in this paper include those that have been repeatedly used in studies of search behaviors over several decades of research. They can be interpreted as the research community's common-sense understanding of what's important about search tasks. Thus, we have some sense of the directions that we need to pursue in terms of improving our understanding of search tasks.

While there is some consensus about which attributes of search tasks are most important, there is no consensus about how those attributes should be defined and operationalized. A task that is categorized as simple by one researcher might be categorized as specific by another. A task that is categorized as a lookup task by one researcher might be categorized as an informational task by another. Furthermore, tasks used to operationalize one variable may have additional attributes not accounted for in the study design, which may confound the results. If we ever want to compare results across studies, we must improve our understanding of the search tasks in use as either control variables or independent variables.

The task attribute that has garnered the most investigation is task complexity. There have been a number of studies that compare simple tasks with those that are more complex, but there are additional task attributes that warrant consideration. These include the topic, domain or subject area of a search task [27, and the informational goal, whether it be to learn about something, make a decision, solve a problem, etc.

# 3. RECOMMENDATIONS FOR DEVELOPING SEARCH TASKS

Ideally, researchers will develop search tasks that are realistic and that appropriately motivate the study subjects to perform a realistic search. Thus, a logical starting point is to elicit real-life situations and task cases from the population of interest [9, 12] or mine existing transaction logs [20]. From these stories of real tasks and the searches they engendered, the researcher can develop search tasks that simulate realistic situations.

Using simulated situations to present search tasks, as defined by Borlund [4, 5], increases the validity of the search tasks by decreasing their artificiality. Borlund suggests that simulated situations be made up of two parts: the simulated work task situation and the indicative request. Byström and Hansen [9] go into even more detail, recommending that three levels of description should be used to specify a search task: a contextual description, a situational description, and a topical description and query. Within the context of a study of search behaviors, the simulated situations are "meant to trigger individual information problems in test persons in a controlled manner" [15, p.284]. The recommendations of these scholars are consistent, in that they all encourage the inclusion of contextual information in the specification of a search task. As search task descriptions become increasingly detailed, it is important to validate the effect of search tasks through pre-testing and/or by collecting user feedback during the study. Given that the construction and validation of robust contextualized tasks is challenging, researchers should consider the option of sharing, reusing and/or customizing existing tasks.

In addition, in order to effectively evaluate interactive IR systems, it will be necessary to develop more complex search tasks and more exploratory search tasks. Many IR researchers want to design systems that support a broader range of activities – both more complex activities and ongoing activities. Thus, we need to understand how to develop search tasks that can support experimental research and system evaluation in more realistic contexts. Two recent articles suggest approaches to this problem. Kules and Capra [20] designed a template for developing exploratory tasks. The situation specified in the template was a search conducted while writing a paper for a class, with the search to be conducted on the university's library catalog. While specific to their study, it's easy to imagine that similar templates might be developed for other situations. Taking a different approach, Li and Belkin [22] identified and developed a taxonomy of the attributes of tasks (including both work tasks and search tasks). The taxonomy includes both generic facets of tasks (e.g., source, time-related attributes, and goals) and common attributes of tasks (e.g., complexity, difficulty, and urgency. As researchers develop search tasks, they can use this taxonomy to specify the task attributes of interest for a particular study.

While we need additional research to fully understand which attributes of tasks have an important effect on search behaviors, we can and should improve the research base now by more carefully documenting the attributes of the search tasks assigned to research subjects. In each paper, the researcher should indicate which attributes of the search tasks are most salient. These attributes should then be defined and their operationalization should be described. For example, a study might investigate the search behaviors associated with different levels of task complexity. The researcher's definition of complexity should be documented, and how different levels of complexity were incorporate into the search tasks should be clearly explained. Only with such clear specification of our research methods can we gain leverage on understanding the effects of search tasks on search behaviors.

To contribute to the improvement of experimental methods for studying search behaviors, we are initiating a project that will compile and analyze studies employing assigned search tasks. From each study, we are capturing the way in which the search tasks are categorized or defined (if they are), as well as the full text of the search tasks themselves. Through this analysis, we hope to be able to more accurately describe the search tasks that have been used in past studies, as well as define the attributes of search tasks which are most likely to influence the outcomes of future studies.

## 4. ACKNOWLEDGMENTS

We thank Anthony Hughes, Jung Sun Oh, Ashraf Farrag, and Chris Doty, each of whom has made significant contributions to the compilation of a database of assigned search tasks, used as the basis for this analysis.

## 5. REFERENCES

[1] Bell, D. & Ruthven, I. (2004). Searcher's Assessments of Task Complexity for Web Searching. *Advances in Information Retrieval, 26th European Conference on IR Research, ECIR 2004*. Springer, 57-71.

[2] Bilal, D. (2001). Children's use of the Yahooligans! Web search engine: II. Cognitive and physical behaviors on research tasks. *J. of the Am. Soc. for Info. Sci. & Tech.*, 52(2), 118-136.

[3] Bilal, D. (2002). Children's use of the Yahooligans! Web search engine. III. Cognitive and physical behaviors on fully self-generated search tasks. *J. of the Am. Soc. for Info. Sci. & Tech., 53*(13), 1170-1183.

[4] Borlund, P. (2003). The IIR evaluation model: a framework for evaluation of interactive information retrieval systems. *Info. Research, 8*(3). http://informationr.net/ir/8-3/paper152.html.

[5] Borlund, P., & Ingwersen, P. (1997). The development of a method for the evaluation of interactive information retrieval systems. *J. of Documentation, 53*(3), 225-250.

[6] Broder, A. (2002). A taxonomy of Web search. *SIGIR Forum, 36*(2), 3-10.

[7] Browne, G. J., Pitts, M. G., & Wetherbe, J. C. (2007). Cognitive stopping rules for terminating information search in online tasks. *MIS Quarterly*, 31(1), 89-104.

[8] Byström, K. (2002). Information and information sources in tasks of varying complexity. *J. of the Am. Soc. for Info. Sci. & Tech., 53*(13), 1170-1183.

[9] Byström, K., & Hansen, P. (2005). Conceptual framework for tasks in information studies. *J. of the Am. Soc. for Info. Sci. & Tech., 56*(10), 1050-1061.

[10] Campbell, D. J. (1988). Task complexity; a review and analysis. *Acad. of Mgmt. Rev.*, 13(1), 40-52.

[11] Capra, R., Marchionini, G., Oh, J. S., Stutzman, F., & Zhang, Y. (2007). Effects of structure and interaction style on distinct search tasks. *JCDL 2007: Proc. of the 7th ACM/IEEE Joint Conference on Digital Libraries,* 442-451.

[12] Elsweiler, D., & Ruthven, I. (2007). Toward task-based personal information management evaluations. *Proc. of the 30th ACM Conference on Research and Development in Information Retrieval (SIGIR '07),* 22-30.

[13] Fidel, R., Davies, R.K., Douglass, M.H., Holder, J.K., Hopkins, C.J., Kushner, E.J., Miyagishima, B.K., & Toney, C.D. (1999). A visit to the information mall: Web searching behavior of high school students. *J. of the Am. Soc. for Info. Sci., 50*(1), 24 –37.

[14] Gwizdka, J., & Spence, I. (2006). What can searching behavior tell us about the difficulty of information tasks? A study of web navigation. *Proc. of the Am. Soc. for Info. Sci. & Tech.* doi:10.1002/meet.14504301167.

[15] Ingwersen, P., & Järvelin, K. (2005). *The Turn: Integration of Information Seeking and Retrieval in Context.* Springer.

[16] Jansen, B.J., Booth, D.L., & Spink, A. (2008). Determining the informational, navigational, and transactional intent of Web queries. *Info. Proc. & Mgmt., 44*(3), 1251-1266.

[17] Joachims, T., Granka, L., Pan, B., Hembrooke, H., Radlinski, F., & Gay, G. (2007). Evaluating the accuracy of implicit feedback from clicks and query reformulations in Web search. *ACM Trans. on Info. Sys., 25*(2), Article 7.

[18] Kelly, D. (2009). Methods for evaluating interactive information retrieval systems with users. *Foundations and Trends in Info. Retrieval, 3*(1-2).

[19] Kim, K.-S., & Allen, B. (2002). Cognitive and task influences on web searching behavior. *J. of the Am. Soc. for Info. Sci. & Tech., 53*(2), 109-119.

[20] Kules, B., & Capra, R. 2008). Creating exploratory tasks for a faceted search interface. *HCIR 2008 Workshop Proceedings,* 18-21.

[21] Lazonder, A. W., Biemans, H. J. A., & Wopereis, I. G. J. H. (2000). Differences between novice and experienced users in searching information on the World Wide Web. *J. of the Am. Soc. for Info. Sci.*, 51(6), 576-581.

[22] Li, Y., & Belkin, N.J. (2008). A faceted approach to conceptualizing tasks in information seeking. *Information Processing & Management, 44*(6), 1822-1837.

[23] Lorigo, L., Pan, B., Hembrooke, H., Joachims, T., Granka, L., & Gay, G. (2006). The influence of task and gender on search and evaluation behavior using Google. *Info. Proc. & Mgmt., 42*(4), 1123-1131.

[24] Marchionini, G. (2006). Exploratory search: From finding to understanding. *Commun. of the ACM, 49*(4), 41-46.

[25] Rouet, J.-F. (2003). What was I looking for? The influence of task specificity and prior knowledge on students' search strategies in hypertext. *Interact. with Comp., 15*(3), 409-428.

[26] Sharit, J., Hernandez, M.A., Czaja, S.J., & Pirolli, P. (2008). Investigating the roles of knowledge and cognitive abilities in older adult information seeking on the Web. *ACM Trans. on Computer-Human Interaction, 15*(1), Article 3.

[27] Toms, E.G., Freund, L., Kopak, R. & Bartlett, J.C. (2003). The effect of task domain on search. *Proc. of CASCON 2003*, 1-9.

[28] Toms, E.G, Kopak, R., Bartlett, J. & Freund, L. (2001). Selecting versus describing: the efficacy of categories in exploring the Web. *Proc. of the 10th Text Retrieval Conference*, 64-70.

[29] Vakkari, P. (1999). Task complexity, problem structure and information actions: Integrating studies on information seeking and retrieval. *Info. Proc. & Mgmt., 35*, 819-837.

[30] Vakkari, P. (2003). Task-based information searching. *Annual Rev. of Info. Sci. & Tech., 37*, 413-463.

[31] White, R.W., Kules, B., Drucker, S.M., & schraefel, m.c. (2006). Introduction [to special issue on Supporting exploratory search]. *Commun. of the ACM, 49*(4), 36-39.

[32] White, R. W., & Marchionini, G. (2007). Examining the effectiveness of real-time query expansion. *Info. Proc. & Mgmt., 43*(3), 685-704.

[33] Wildemuth, B. M., & Hughes, A. (2005). Perspectives on the tasks in which information behaviors are embedded. In Fisher, K E., Erdelez, S., & McKechnie, L. (Eds.), *Theories of Information Behavior*. Medford, NJ: Information Today, for ASIST, 275-279.

[34] Woodruff, A., Rosenholtz, R., Morrison, J. B., Faulring, A., & Pirolli, P. (2002). A comparison of the use of text summaries, plain thumbnails, and enhanced thumbnails for web search tasks. *J. of the Am. Soc. for Info. Sci. & Tech., 53*(2), 172-185.

# Visual Interaction for Personalized Information Retrieval

Jae-wook Ahn
School of Information Sciences
University of Pittsburgh
Pittsburgh, PA 15260
jahn@mail.sis.pitt.edu

Peter Brusilovsky
School of Information Sciences
University of Pittsburgh
Pittsburgh, PA 15260
peterb@mail.sis.pitt.edu

## ABSTRACT

There are two promising answers to the classic information overload problem: (1) personalized search and (2) exploratory search. Personalized search stresses more on the algorithmic side and the exploratory search pays more attention on the user interface to help users to achieve better search results. We believe that by combining these two approaches, we can provide users with a better solution than the old ranked list based interaction mechanism of personalized search. This paper proposes to incorporate interactive visualization into personalized search. We extended a well known visualization method called VIBE (Visual Information Browsing Environment) to visualize user models and then incorporated it into the personalized search framework. We expect our approach will be able to help users to better explore the information space and locate relevant information more efficiently. Here, we showed the concept and the potential of this adaptive visualization method. We also introduced a new prototypical personalized search system and its interaction model implementing the adaptive visualization idea.

## 1. INTRODUCTION

The information overload problem has been one of the major motivations of the information retrieval community. A lot of effective methodologies and algorithms have been designed to help users to locate relevant information from the massive pool of candidate documents. Yet the Web constantly pose new challenges to the researchers. Nowadays, when the Web emerged as a universal source of information, a range of problems, which users attempt to solve using the Web expanded as well. More and more frequently people use the Web for exploratory search tasks, which required multiple searches and complex processing of information. Since search engines or traditional search tools provided by industry are found not to be adequate to support exploratory search, researchers explored a range of alternative solutions such as (1) personalized search [5] and (2) exploratory search interfaces (or interactive search) [4]. Personalized search applies artificial intelligence to attract user attention to most relevant results, while exploratory search interfaces attempt to empower human's own intelligence by providing more interactive and expressive user interfaces so that they can reach better search results.

While these approaches are typically considered as alternatives, they are really complementary. We believe that both personalized search and exploratory search interfaces can benefit from each other. In particular, most of personalized searching algorithms are still relying on query-ranked list model. Despite the simplicity and straightforwardness of this model, we still see the potential of a more advanced user interface with which users can better understand what is happening inside the search engine and eventually contribute to improving the search results. We expect to improve the performance and the transparency of personalized IR systems by providing a more dynamic and flexible interaction model between the system and the users, which is typical for exploratory search interfaces.

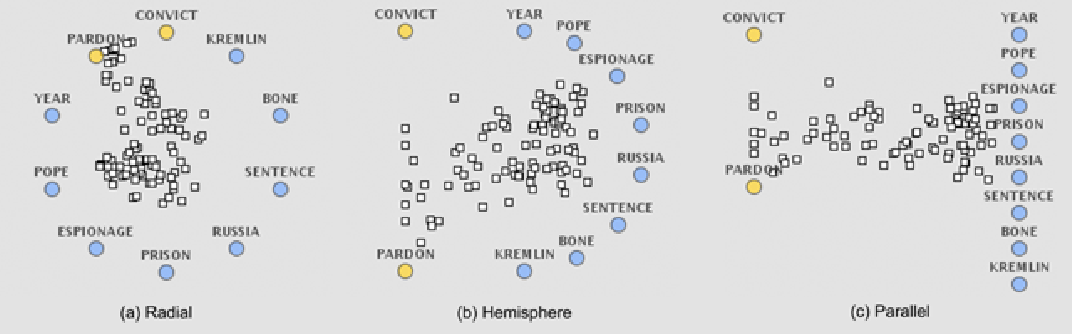
We propose to incorporate interactive visualization into personalized search in order to achieve this goal. By combining the personalized search and the interactive visualization, we expect our approach will be able to help users to better explore the information space and locate relevant information more efficiently. We extended a well-known visualization framework called VIBE (Visual Information Browsing Environment) [6] so that it can visualize the user models and the personalized search results. Beyond the past studies [1, 2] that have found some potentials of this adaptive visualization method, we are planning to conduct a full-scale user study to investigate its strengths and weaknesses. This paper introduces the visualization-based personalized IR idea (section 2) and the preliminary experiment results (section 3), and then describes the details of our prototypical personalized search system (section 4) and its interaction model constructed for the future user study.

## 2. VISUALIZING USER MODELS FOR PERSONALIZED SEARCH

### 2.1 VIBE – the foundation visualization model

VIBE (Visual Information Browsing Environment) is a relevance-based visualization method developed at Molde College and the University of Pittsburgh [6]. It displays a number of documents and determines their positions by their similarity ratios to a group of reference points called POIs (Point of Interest). Therefore, if a document has a similarity score 0.2 to POI$_a$ and 0.8 to POI$_b$, then it is placed on a position four times closer to POI$_b$ due to its four times bigger similarity value. It can visualize more than just two POIs, so that users can investigate the relationships among multiple reference points (POIs) and the documents (See Figure 1 and 2 as examples).

The most general idea to apply the VIBE visualization to IR may be defining the query terms as POIs and visualize them with the retrieved documents. However, all the POIs are treated equally in terms of their dimensions in this idea. This is a shortcoming when we consider multiple *groups* of POIs that represent different layers of meanings. For example, if we want to define POIs that represent user interests estimated by the system, we need to differentiate them from the queries directly entered by the users. However, traditional VIBE cannot handle this difference. Therefore, we raised a simple idea that can overcome this shortcoming – by spatially and visually separating two POI groups.



(a) Radial (b) Hemisphere (c) Parallel

**Figure 1: Adaptive VIBE layouts – showing three layouts. Yellow (*CONVICT* and *PARDON*) and blue (*YEAR*, *POPE*, and so on) POIs are query terms and user model keywords respectively. White squares are retrieved documents.**

## 2.2 Adaptive VIBE – extension of VIBE for user model visualization

VIBE-based IR visualization approaches usually used a *radial* layout for their initial placement of POIs (Figure 1a). Like a round table, all POIs are equal in this idea. However, in our application, where we want to visualize the personalized search result, we need to discriminate the documents retrieved by the user model and by the user queries. We wanted to let users know which documents were more affected by the user models than the query terms, or vice versa. We could achieve this discrimination by separating the groups of POIs spatially on the screen. Figure 1 shows an example. In addition to the traditional layout Radial (a), we defined two new layouts called Hemisphere (b) and Parallel (c). The two latter layouts try to locate two groups of POIs – query and user model POIs – in horizontally separate places and then break up the spaces between two groups. By this POI group separation, the documents under the effect of each group are separated spatially again and the users can easily investigate each cluster of documents to find out relevant information. The difference between the two are whether the separation is bigger (Parallel) or moderate (Hemisphere). In addition to the location, we could use different color codes for two groups. In the figure, the query terms and the user model keywords were painted in yellow and blue respectively.

The Radial layout (a) is identical with the non-adaptive VIBE except the color coding. By comparing it with (b) or (c), we can understand the advantage of Adaptive VIBE. The documents are mostly cluttered around the center and it is hard to see the aboutness of the documents to the POIs, especially hard to catch whether they are more about the query or more about the user model. However, in the layout (b) and (c) we can clearly separate the document groups by whether they are closer (in terms of location as well as meaning) to the query (yellow POIs) or the user model (blue POIs). In the Hemisphere layout (b), there still remains a bit of the clutter as in the Radial layout whereas the clutter is almost disappeared in the Parallel layout (c). However, we can still see the advantage of the Hemisphere layout, because it is easier to find the difference *within* each POI group (user model or query POIs). For example, it is easier to find out documents closer to *POPE* than *RUSSIA* in the Hemisphere rather than in the Parallel layout.

# 3. PRELIMINARY EXPERIMENTS

In order to test if the idea described in the previous section can really benefit the personalized IR, we conducted an experiment [2]. The experiment was feasible with an existing dataset generated from our previous text-based personalized search study [3]. We had built a task-based personalized IR system and conducted a user study, where the subjects were asked to search for a specific information while the system built user models and tried to help users to find relevant information using the user models. Because all the activities of the users and the system were recorded into the log files, we could extract the core information of the search sessions such as (1) user query (2) retrieved documents (with similarity scores) (3) user model content, and then recreate the Adaptive VIBE visualization replacing the textual ranked lists.

After recreating the visualization with the real data extracted from the previous study, we were able to check if the simulated visualization would be able to help the searcher. Figure 2 shows one snapshot of the images generated in the experiment. Because we were equipped with the tasks to be solved by the subjects and the groundtruth of the topics from the user study, we were able to mark the relevant documents to the given topic and examine how their spatial distributions look like. As can be seen in the figure, there tended to appear clusters of relevant (green) and irrelevant (red) documents. What is more interesting is, the relevant document clusters were closer to the user model side (right hand side according to our layout definitions) than the query side. This trend became stronger as we used the Hemisphere, and then the Parallel layouts. We were able to check this finding numerically by examining the centroid positions of the (ir)relevant document clusters. Table 1 compares the $x$-coordinates of the cluster centroids by the three layouts. It shows that the $x$-coordinates of the relevant documents were greater than those of the irrelevant document clusters, and therefore they were closer to the user model because the user models were placed in the right hand side of the screen. The distances between the centroids grew bigger in adaptive visualization layouts (42.40 and 92.94 pixels with Hemisphere and Parallel respectively) than the non-adaptive one (20.4 pixels). For more details about the experiment and additional analysis, please check [2].

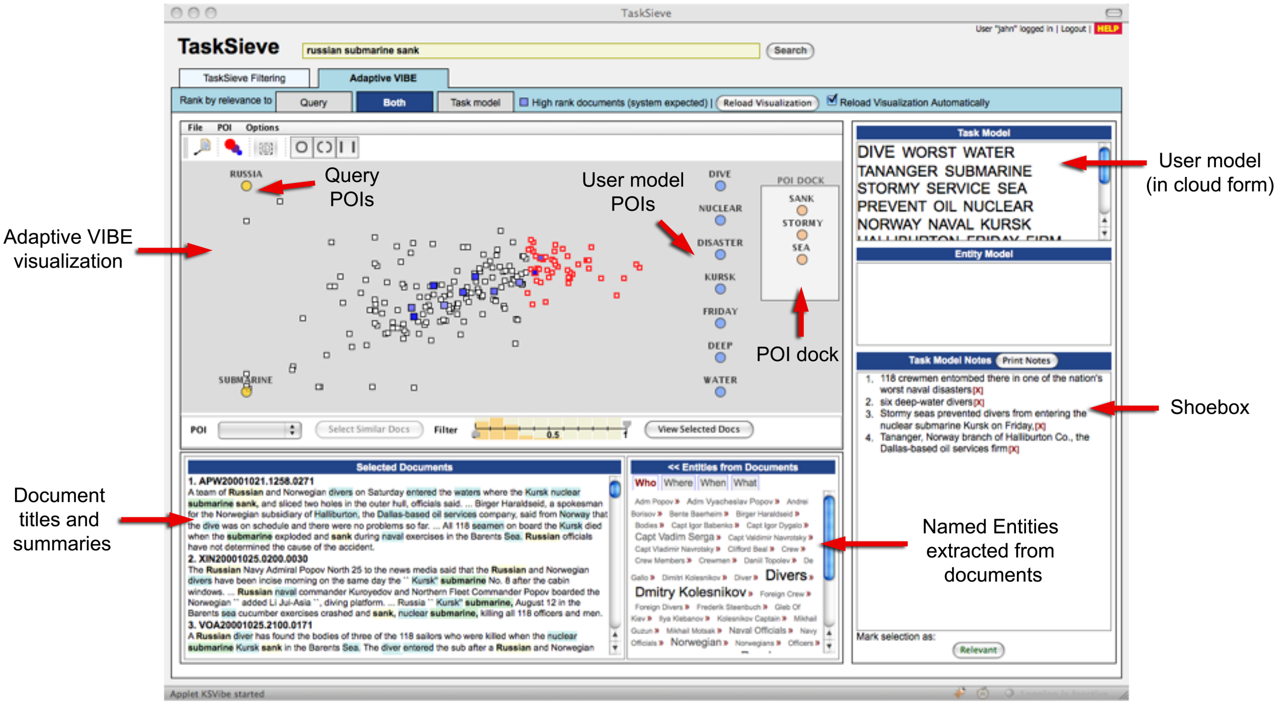
**Table 1: Comparison of relevant and irrelevant document cluster centroid positions on the visualization (averages in pixel)**

| | Radial | Hemisphere | Parallel |
|---|---|---|---|
| Relevant | 304.3 | 337.7 | 300.9 |
| Non-relevant | 283.9 | 295.3 | 207.96 |
| Difference<br>(relevant – irrelevant) | 20.4 | 42.40 | 92.94 |

# 4. A PROTOTYPE SYSTEM

## 4.1 Adaptive Visualization

Encouraged by the preliminary experiment result, we decided to incorporate Adaptive VIBE in a real personalized search system.



1. APW20001021.1258.0271
A team of Russian and Norwegian divers on Saturday entered the waters where the Kursk nuclear submarine sank, and sliced two holes in the outer hull, officials said. ... Birger Haraldseid, a spokesman for the Norwegian subsidiary of Halliburton, the Dallas-based oil services company, said from Norway that the dive was on schedule and there were no problems so far. ... All 118 seamen on board the Kursk died when the submarine exploded and sank during naval exercises in the Barents Sea. Russian officials have not determined the cause of the accident.

2. XIN20001025.0200.0030
The Russian Navy Admiral Popov North 25 to the news media said that the Russian and Norwegian divers have been incise morning on the same day the `` Kursk'' submarine No. 8 after the cabin windows. ... Russian naval commander Kuroyedov and Northern Fleet Commander Popov boarded the Norwegian '' added Li Jui-Asia '', diving platform. ... Russia '' Kursk'' submarine, August 12 in the Barents sea cucumber exercises crashed and sank, nuclear submarine, killing all 118 officers and men.

3. VOA20001025.2100.0171
A Russian diver has found the bodies of three of the 118 sailors who were killed when the nuclear submarine Kursk sank in the Barents Sea. The diver entered the sub after a Russian and Norwegian

Adm Popov » Adm Vyacheslav Popov » Andrei Borisov » Bente Baerheim » Birger Haraldseid » Bodies » Capt Igor Babenko » Capt Igor Dygalo » Capt Vadim Serga » Capt Valdimir Navrotsky » Capt Vladimir Navrotsky » Clifford Beal » Crew » Crew Members » Crewmen » Danil Topolev » De Gallo » Dimitri Kolesnikov » Diver » Divers » Dmitry Kolesnikov » Foreign Crew » Foreign Divers » Frederik Steenbuch » Gleb Of Kiev » Ilya Klebanov » Kolesnikov Captain » Mikhail Guzun » Mikhail Motsak » Naval Officials » Navy Officials » Norwegian » Norwegians » Officers »

1. 118 crewmen entombed there in one of the nation's worst naval disasters[x]
2. six deep-water divers[x]
3. Stormy seas prevented divers from entering the nuclear submarine Kursk on Friday,[x]
4. Tananger, Norway branch of Halliburton Co., the Dallas-based oil services firm[x]

**Figure 3: Prototype System User Interface – Visual TaskSieve (TaskSieve plus Adaptive VIBE)**

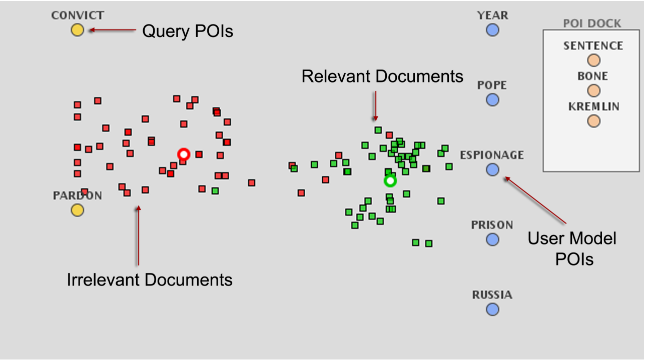


**Figure 2: Adaptive VIBE experiment – green and red squares mean relevant/irrelevant documents.**

Figure 3 is a screenshot of a prototypical personalized search system called TaskSieve [3] empowered by Adaptive VIBE. TaskSieve is a personalized search system with which the users can find out relevant information following the information foraging and sensemaking process [7]. The users initiates searching by entering queries but their search process gets enriched by the user models which are constructed from the information foraging and the sensemaking loops. The users can annotate interesting (relevant or informative) evidences (or *notes*) they found into a special place called *shoebox* (lower right corner of the screen). Their user models are extracted from the content of the shoebox instantly and automatically, shown in the form of *cloud* (upper right corner) and applied to the personalized search engine on the fly to update the list of retrieved documents.

Adding Adaptive VIBE into TaskSieve (Visual TaskSieve), the old textual ranked list was replaced with the Adaptive VIBE visualization as shown in Figure 3 (occupying the biggest real estate of the screen). User query and the user model is provided as POIs, and the POI layouts described in section 2 are used to make the visualization adaptive. The interaction models of the text-based TaskSieve (a) and Visual TaskSieve (b) are compared in Figure 4. Two models are almost identical except Visual TaskSieve makes use of the visualization (green box) for representing the search results. In traditional IR models (including old TaskSieve), users examine the list of documents, their ranks, navigate across pages, and then find out relevant documents to improve the initial search results. In the Visual TaskSieve model, however, we present the retrieved documents visually and users examine them spatially and interactively. The following sections describe the spatial interaction model of Visual TaskSieve.

## 4.2 Spatial User Interaction on Retrieval Results

The evolution from the textual ranked list to the spatial visualization changes the way of user interaction too. Users are provided with very limited information about the retrieved documents from the ranked lists, their vertical positions in the list. However, spatial visualizations such as VIBE and Adaptive VIBE added more dimensions and users can learn the aboutness of the documents to the reference points (POI) and their search context (personalization). As the spatial distribution of the retrieved documents became important, we added a couple of tools to better support the spatial examination of the documents and feedbacks.

***Spatial filtering of retrieved documents*** – Users can select (or filter) documents spatially on the visualization using a marquee tool. Detailed information such as titles, surrogates, and the links to the fulltext of the selected documents are displayed. The red-colored documents (squares) in Figure 3 are the examples of the selected documents using the marquee tool and their information is shown below the visualization. Because spatial distribution of the documents is tightly connected to the aboutness of the documents

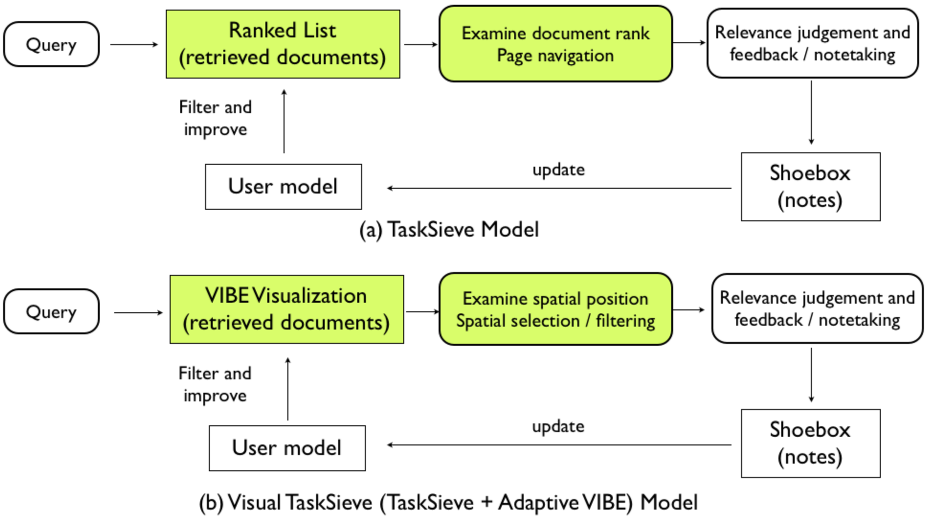


**Figure 4: Comparing the interaction models of TaskSieve and Visual TaskSieve**

(more about UM, query, or a specific POI), this selection tool can let users more promptly learn the contexts of the retrieved documents.

***Linking POI and document filtering by a double slider*** – Users can directly point a POI and filter related documents to the POI (and select/see the detailed information of the documents). They can specify the lower and upper similarity threshold of the documents to the POI using a double slider (Figure 5). This function plays a similar role with the marquee selection, but is expected to help users to select documents more precisely in a reverse direction, from a POI to documents.

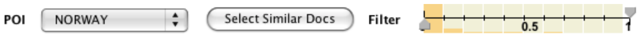


**Figure 5: Selecting documents from a POI: users can set the low and high similarity thresholds of the documents to be selected using the double slider**

***POI dock*** – Sometimes, too many POIs at the same time on the screen can make users feel lost and frustrated. However, we cannot just limit the amount of information that users can access. Therefore, we added a special area where some POIs are temporarily moved out so that their effects on the visualization are disabled. We call it as a POI dock, which is shown in Figure 3 next to the user model POIs. Users can drag POIs to/from this area and interactively examine the effect of each POI they are interested.

### 4.3 Concept Level User Modeling and POI Extraction

In our previous studies, the elements of the user models were always keywords (or terms). Even though the keyword-based user models were working relatively good, we expect concept-based user models would help users better than the simple keywords. Therefore, we are devising ideas to realize the concept-based user models in our current Visual TaskSieve system. We can extract named-entities from the retrieved (or selected) documents and use them as user model elements, and show them as POIs on the visualization. In Figure 4, the user can examine the NEs from the retrieved documents in the cloud form (below middle), and then they can pick and add some of them as POIs to the visualization.

## 5. CONCLUSIONS

This paper showed our ongoing effort to incorporate adaptive visualization into personalized search. We have extended a well known VIBE visualization algorithm for adaptive visualization and have investigated its effectiveness. We are constructing a full-fledged prototype personalized IR system that supports the adaptive VIBE visualization. We showed the details of the idea and the prototype as well as the preliminary experimental results. We are planning to finish the construction of the system soon and conduct a full-scale user study in order to learn the strengths and the shortcomings of our visual IR idea.

## 6. REFERENCES

[1] J.-w. Ahn and P. Brusilovsky. From user query to user model and back: Adaptive relevance-based visualization for information foraging. In *WI '07: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, pages 706–712, 2007.

[2] J.-w. Ahn and P. Brusilovsky. Adaptive visualization of search results: Bringing user models to visual analytics. *Information Visualization*, 8(3):167–179, 2009.

[3] J.-w. Ahn, P. Brusilovsky, D. He, J. Grady, and Q. Li. Personalized web exploration with task models. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 1–10, New York, NY, USA, 2008. ACM.

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

[5] A. Micarelli, F. Gasparetti, F. Sciarrone, and S. Gauch. Personalized search on the world wide web. pages 195–230. 2007.

[6] K. A. Olsen, R. R. Korfhage, K. M. Sochats, M. B. Spring, and J. G. Williams. Visualization of a document collection: the vibe system. *Inf. Process. Manage.*, 29(1):69–81, 1993.

[7] P. Pirolli and S. Card. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proceedings of International Conference on Intelligence Analysis*, 2005.

# PuppyIR: Designing an Open Source Framework for Interactive Information Services for Children

Leif Azzopardi, Richard Glassey,
Mounia Lalmas, Tamara Polajnar
Dept. Computing Science
University of Glasgow
Scotland, U.K.
{leif, rjg, tamara, mounia}@dcs.gla.ac.uk

Ian Ruthven
Dept. of Computer and Information Sciences
University of Strathclyde
Scotland, U.K.
ir@cis.strath.ac.uk

## ABSTRACT

One of the main aims of the PuppyIR project is to provide an open source framework for the development of Interactive Information Retrieval Services. The main focus of the project is directed towards developing such services for children, which introduces a number of novel and challenging issues to address (such as language development, security, moderation, etc).

In this poster paper, we outline the preliminary high-level design of the open source framework. The framework uses a layered architecture to minimize dependencies between the user-side concerns of interaction and presentation, and the system-side concerns of aggregating content from multiple sources and processing information appropriately. Each layer will consist of a series of interchangeable components, which can be interconnected to form a complete service. To facilitate the construction of diverse information services, a dataflow language is proposed to enable the assembly of the components in an intuitive and visual manner. One of the the design goals of the architecture, and ultimate measures of success, is to provide a "lego" style building block environment in which researchers and developers of any age can build their own information service.

This poster provides the starting point for the design of the framework and aims to seek comments, feedback and suggestions from the community in order to improve and refine the architecture.

## 1. INTRODUCTION

Today children are exploring the Internet from a very early age. However, much of the content and services available on the Internet have been designed for adults. And so while many children actually possess better computing skills than their parents, they are also left vulnerable to the darker side of the net (in the worst case) or overwhelmed and overloaded when searching for information available online [7]. Furthermore, they also have to interact and engage with systems and tools that are not specifically designed to consider or develop their cognitive, emotional and intellectual abilities [3].

Consequently, many children experience difficulties during information seeking, as they have to contend with a number of challenges: non-intuitive search interfaces; unmoderated and unverified content; language and understanding issues; and structured and prescribed interaction styles. In order to help children effectively, efficiently and enjoyably engage with information, entertainment, friends, and so forth online, it is important to provide access to such information services in ways that are consistent with their learning, cognitive development and curriculum [3]. To support the investigation of such challenges, the PuppyIR project aims to provide the infrastructure to construct and build interactive information services using an open source framework.

The remainder of the poster is as follows: In the next section we discuss the aims of the PuppyIR project, then we outline the high-level design of the open source framework, before providing an example of constructing an information service. Finally, we conclude with a brief summary.

## 2. PUPPYIR PROJECT

PuppyIR[1] is an European Union project that is funded under the European Community's Seventh Framework Programme. The project will run over the next three years and involves seven partners from within European Union. The aims of the project are as follows:

1. Investigation and promotion of new interaction paradigms for children's access systems.

2. A number of interrelated research questions will be addressed that contribute to the interaction of children with information services. The focus will be on presentation, mining and language issues for the appropriate development of these information services.

3. An open source framework will be created to support the development of common interchangeable components to support the realization of information services for children.

4. Identification of appropriate evaluation measures and develops an evaluation framework for information services for children.

*HCIR* '09 Washington, USA

[1] `http://www.puppyir.eu/`

In this poster paper, we shall focus on the 3rd aim of this ambitious and exciting project and outline the preliminary design for the high level architecture of the PuppyIR Open Source Framework, as a way to open up the project to the wider community and seek input and feedback on the design and architecture.

## 3. OPEN SOURCE FRAMEWORK DESIGN

One of the main design goals of this project is to create an open source framework that supports the rapid development of child-friendly Interactive Information Retrieval (IIR) services. The provision of such a framework enables both system developers and researchers to design and create different types of IIR systems using a common library of components and configurations, and to do so quickly and cost-effectively.

With that in mind, the PuppyIR Open Source Framework (PIR-OSF) aims to deliver an open, reconfigurable and reusable software framework. It avoids unnecessary duplication of effort and allows for rapid development of multiple independent IIR services. Finally it provides an effective evaluation environment for ongoing HCI and IR research, based around an open and common toolset.

To achieve these aims, the design of PIR-OSF establishes three fundamental abstractions for modeling a generic IIR service:

1. **Layered Architecture:** provides the separation of concern between the common aspects of an IIR service (information sources, information processing, visualization and interaction) that enables multiple environments and application scenarios to be addressed using a single framework.

2. **Rich Information Objects:** a flexible abstraction for any type of information content (text, images, audio, movie, mixed, etc.) and its accompanying metadata required by an IIR service.

3. **Dataflow Language for Information Processing:** allow both users and developers to reuse, visually reconfigure alternative information processing networks to deliver the most appropriate information.

The following sections discuss these abstractions and their implications upon the design of the PIR-OSF.

### 3.1 Layered Architecture

In its most basic form, an IIR service allows users to interact by submitting their queries, then presenting them with the static results (e.g. via a search engine) or a stream of information (e.g. a RSS feed) using an appropriate form of presentation. The architecture of the PIR-OSF represents this high-level system view using a series of independent, stacked layers.

This design decision reflects the different requirements of building information services for multiple environments and application scenarios. More fundamentally, it recognizes the close relationship that exists between human-computer interaction and information retrieval. By providing a common framework, researchers from both fields can conduct research within their appropriate layer(s) and reuse components from a pre-existing library built by others. Thus, the entry barrier to building a complete system is dramatically lowered, reducing the effort for both systems' designers and the HCIR research communities.

A concrete illustration of the importance of a layered architecture is to consider a unified museum guide for children. Results of a query issued at a public static terminal or from their own mobile device could be projected onto a nearby wall-mounted display, taking advantage of the extra screen space and perhaps touch-oriented interaction. Should no display be nearby, the user could still use their mobile device, with the same underlying system, but the presentation of and interaction with the results would be vastly different. To accommodate these differences, a flexible architecture is essential. Figure 1 shows the layered architecture of PIR-OSF. A conceptual divide separates the lower system-side layers from the upper user-side layers.

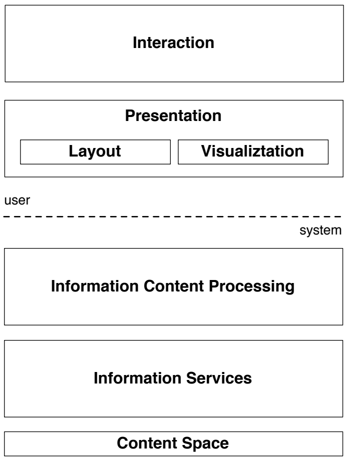


**Figure 1: Layered architecture of the PuppyIR Open Source Framework**

**System-side Layers:** At the bottom of the stack, the Content Space layer represents all of the potential information sources that are indexed and made available to the rest of a system. The Information Services layer encompasses the various interfaces to online services, such as Google, Yahoo, Bing, Youtube, Flickr, Twitter (uncooperative services) and so on, as well as content indexed by offline services such as Lemur, Lucene and Terrier (cooperative services). This layer is responsible for using the respective API of each service and specifies the interface for creating result wrappers for each service, such that results can be returned as collections or streams of Rich Information Objects (RIO).

The Information Content Processing layer manipulates the RIOs by passing them through a network of linked components called Information Content Processors (ICP), configured using a dataflow-oriented approach. For example, a typical ICP may moderate the content of a RIO and return the modified result for further processing. More complex ICPs may act as aggregators and distributors of multiple RIOs, as shown in Fig. 2. Alternatively, two summarizer ICPs could be evaluated by swapping them into the same ICP network, without changing any other aspect of the configuration. This networked aspect provides complete flexibility with the configuration, making it suitable to develop a wide range of systems.

**User-side Layers:** The upper layers relate to the user-

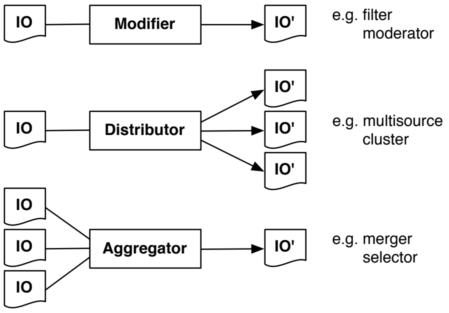


**Figure 2: Information Content Processor (ICP) configurations**

side concerns; how RIOs are presented, and how the user can interact with the system in terms of query formulation and interactive feedback. The Presentation layer deals with the overall layout of containers of RIOs (e.g. a webpage that has several blocks for separating image and textual results) and the visualization of RIOs within the containers (e.g. a ranked list of results or a carousel of images). The Interaction layer corresponds to the variety of modes and modalities that a user may utilize when using the system, ranging from the familiar textual and graphical interfaces, onwards to more advanced speech, touch and tangible forms of interaction.

## 3.2 Rich Information Objects

The RIO is an abstraction for all kinds of information that an information service may produce or need to consume. It exposes a public interface that allows the components of PIR-OSF to inspect, manipulate, and exchange it with other components. This abstraction also permits the creation of new content types without needing to modify existing components of the PIR-OSF. For example, new RIOs could be created to mix multiple types of information together as required by a service.

Besides representing information, such as text, images and movies (or combinations), each RIO contains metadata, which either has been extracted from the information (e.g. the title of a web page), or added by a component of PIR-OSF as part of its processing (e.g. a readability rating). The range of metadata is not fixed and is fully extensible depending upon the system needs.

This approach is crucial to providing a common object that all components can inspect, annotate, transform and exchange. Whilst this type of information wrapping inevitably creates overhead costs in terms of time and space when processing objects, it is crucial for the final systems abstraction of PIR-OSF: the dataflow oriented approach to processing RIOs.

## 3.3 Dataflow Language for Information Content Processing

Each service built using the PIR-OSF will have different information processing needs. Since there are a lot of differences between users and their specific needs, as they are children (i.e. lots of variation in age, language skills and language, knowledge, tastes, moods, etc), then the flexibility to configure the information services is paramount. Example information content processing units that will be provided to support the information seeking of children will include: moderation components which filter the content ensuring it is suitable and/or appropriate; summarisation components which will re-factor content to suit the child's reading ability, classification components to aggregate and facet information depending on their context. Consequently, the PuppyIR project will focus on developing information processing components which are designed for children.

The ability to chain together different ICPs is therefore critical towards meeting the different IIR service requirements. One approach might be to treat the chaining of ICPs as a pipe and filter architecture [6], however this reduces the configuration of ICPs to a single, serial pipeline and does not allow a rich network of ICP components to be assembled. Instead, PIR-OSF adopts a dataflow language oriented-approach.

The essence of dataflow programming languages is that a program can be expressed as a directed graph, with nodes acting as primitive instructions (such as arithmetic and comparative operations) and directed arcs representing the data dependencies between these instructions [5] (see Fig. 3).

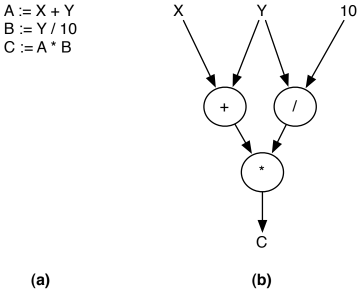


**Figure 3: A simple program (a) and its dataflow equivalent (b)**

Historically, dataflow languages have been closely linked to visual programming, however the infancy of display and interaction technology hampered their development [4]. With advances in user interfaces, increasingly visual approaches are being adopted for dataflow programming, e.g. the MIT Scratch programming environment[2], Apple's Quartz Composer[3] for graphics programming and Yahoo! Pipes for mashing web data together[4]. We adopt this approach for the PIR-OSF, where ICPs replace the instructions and RIOs replace the data. There are three main justifications for using this design choice:

**Visualization:** Early in the history of dataflow programming, it was recognized that dataflow languages could be easily adopted by novice users in order to communicate and construct programs [1]. Given the intuitiveness of such languages, we envisage that not only could systems designers easily (re)-configure the system using visual tools, but the users themselves (in this case children) could become the designers of their own information services; allowing them to better fit their own individual information seeking behavior.

**Parallelism:** Dataflow programming was proposed as a

[2] `http://scratch.mit.edu`

[3] `http://developer.apple.com/graphicsimaging/quartz`

[4] `http://pipes.yahoo.com/pipes`

solution to the von Neumann execution model, in which an instruction can only be executed when the program counter reaches it, even if it could be executed earlier. Many scenarios exist where multiple sources of information can be accessed in parallel and concurrently processed without affecting the final results. Given the importance of responsiveness to the user experience [2], parallelism should be exploited.

**Verifiability:** As the development of any concurrent process tends to be complicated, the use of dataflow programming may help reduce this effort. Networks of ICPs, and the flow of RIOs through them, can be formally modeled, simulated and verified, before spending time and effort trying to develop, debug and maintain a complex concurrent system.

## 4. DESIGNING AN INTERACTIVE INFORMATION SERVICE WITH PIR-OSF

By way of illustration, we present the high-level design of an IIR service using PIR-OSF. We focus mainly upon the component interaction and the flow of information processing for the system-side aspects of the IIR service. The user-side aspects of the system, presentation and interaction, can be assumed to take the form of a web page with aggregated results (a page with a block reserved for web search results, and another for combined image and video results).

Figure 4 shows the IIR service, with two independent ICP networks configured, and the layered architecture overlaid for reference. On the left-hand side, a simple single-path network has been created. In it, a query is submitted, moderated, then sent to the Google search engine using its API. The results returned are wrapped into a set of RIOs at the Information Services layer, before being passed through the moderation and summarizer ICPs. This returns a modified result set (RIO') that can then be presented to the user.

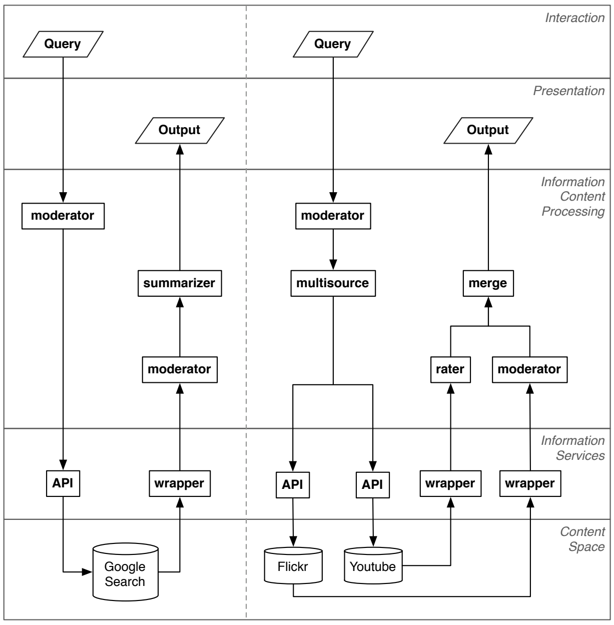


**Figure 4: High-level design of a single IIR service with two independent network configurations of ICPs**

In contrast, the right-hand side of Fig. 4 shows a more complex network of ICPs. The query is passed through a moderator ICP and a multi-source ICP, which can be configured to distribute its input to multiple recipients (see distributor ICP in Fig. 2). In this case, it submits the same query to both Flickr and Youtube. This returns two result sets, which are processed in parallel (one moderated for adult content, whilst the other is ordered by ratings), then merged together into a single result set for presentation to the user. As the same query can be injected into both ICP networks, output can be presented together to the user, in the most appropriate form for the IIR service.

In this example trivial combinations of ICPs were used; more complex networks could be created for specific IIR services. By using a dataflow-oriented approach, IIR services can more easily be reconfigured and customised by both users and systems developers. Furthermore, this approach permits the evaluation of similar ICPs (e.g. two summarizers), by swapping them into an active network, whilst monitoring the user experience and response.

## 5. SUMMARY

In this poster paper, we have outlined the preliminary high-level design of the PuppyIR Open Source Framework for constructing information services. Key to the design is the layered architecture used to assemble the desired service functionality. This is underpinned by a dataflow language that models the flow of Rich Information Objects through a flexible network of Information Content Processors, from the content space to the presentation layer. It is envisaged that this plug and play architecture will enable the rapid development of information services, whilst also being highly extensible and configurable.

**Acknowledgements:** This research is funded by the European Community's Seventh Framework Programme FP7/ 2007-2013 under grant agreement no. 231507.

## 6. REFERENCES

[1] A. L. Davies. Data-driven nets - a class of maximally parallel, output-functional program schemata. Technical report, Burroughs, San Diego, CA, 1974.

[2] W. O. Galitz. *The essential guide to user interface design: an introduction to GUI design principles and techniques.* John Wiley and Sons, 2002.

[3] H. E. Jochmann-Mannak, T. W. C. Huibers, and T. J. M. Sanders. Children's information retrieval: beyond examining search strategies and interface. In *The 2nd BCS-IRSG Symposium: Future Directions in Information Access*, 2008.

[4] W. M. Johnston, J. R. Paul-Hanna, and R. J. Millar. Advances in dataflow programming languages. *ACM Computing Surveys*, 36(1), March 2004.

[5] P. Kosinki. A data flow language for operating systems programming. In *Proceedings of the ACM SIGPLAN–SIGOPS Interface Meeting*, 1973.

[6] R. N. Taylor, N. Medvidovic, and E. Dashofy. *Software Architecture: Foundations, Theory, and Practice.* John Wiley and Sons, 2009.

[7] Y. Zhang. How children find information on the internet: An empirical study and its implications. In *The 3rd Annual Research Conference of the Research Center for Educational Technology (RCET)*, Kent, OH., 2002.

# An Interactive Automatic Document Classification Prototype[†]

Kirk Baker
Collexis
Bethesda, MD 20817
baker@collexis.com

Archna Bhandari
Office of Knowledge Management and Portfolio Analysis
National Institutes of Health
Bethesda, MD 20892

Rao Thotakura
Division of Information Services
National Institutes of Health
Bethesda, MD 20817

[†]Disclaimer: This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any of its employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial products, process, or service by trade name, trade mark, manufacturer, or otherwise, does not constitute or imply its endorsement, recommendation, or favoring by the United States Government. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government and shall not be used for advertising or product endorsement purposes.

## ABSTRACT

In this paper we report on a series of completed and ongoing experiments that involve the integration of fully automatic document classification techniques into an existing manually-oriented document retrieval system. We take our primary findings as positions on the design of an interactive document classifier and retrieval tool.

## General Terms

Design, Experimentation, Human Factors.

## Keywords

Information retrieval, machine learning, user interaction.

## 1. INTRODUCTION

In this paper we report on a series of completed and ongoing experiments that involve the integration of fully automatic document classification techniques into an existing manually-oriented document retrieval system. We take our primary findings as positions on the design of an interactive document classifier and retrieval tool.

The rest of this paper is split into three sections. Section 2 contains background information and a brief technical overview of a document classification system used to report funding allocations across a range of research categories. Section 3 outlines some of the formal system and user requirements that guided our integration of automatic classification techniques into an existing document classification system. In Section 4 we describe our positions on the design of an interactive document classification tool in terms of a functioning prototype that meets these requirements.

## 2. The Research, Condition and Disease Categorization (RCDC) Initiative

In 2006, the United States Congress mandated that the National Institutes of Health (NIH) establish a standardized, automated system for reporting its financial allocations to supported research areas, conditions, and disease categories. The RCDC system is the implementation of this mandate.

It is implemented atop a vector space document retrieval model. Documents are preprocessed by a natural language processing module that extracts only those variations of term strings that correspond to concepts in the RCDC thesaurus. The RCDC thesaurus is a controlled medical ontology that draws from the Medical Subject Headings (MeSH)[1] and CRISP[2] databases, the National Cancer Institute thesaurus[3] , the UMLS Metathesaurus[4], and Jablonski's Dictionary of Medical Acronyms and Abbreviations [1]. Documents are represented as weighted concept vectors where a frequency-based weighting scheme is applied to concept counts and then normalized such that all concept weights fall between 0 and 1.

Definitions of funding areas are also represented as weighted concept vectors along with a similarity threshold, where the set of concepts and weights are determined by subject matter experts and refined in conjunction with ongoing reviews of the set of documents that are retrieved. Concept weights in definitions of funding categories range from -1 to 1, or may be designated as mandatory or excluded. From an information retrieval standpoint, funding area definitions are treated as query vectors and a measure of similarity between each query and each document vector is calculated. When the computed similarity is above threshold for a given query and document, that document is classified as belonging to the given funding category.

The impetus for incorporating automatic document classification techniques into the RCDC system originated in response to the following specifications:

1. a need to alleviate the manual effort required in developing and maintaining category definitions;

[1] http://www.nlm.nih.gov/mesh

[2] http://crisp.cit.nih.gov/crisp/CRISP_Help.help

[3] http://ncit.nci.nih.gov

[4] http://www.nlm.nih.gov/pubs/factsheets/umlsmeta.html

2. a desire to improve classification accuracy of selected existing categories;

3. the ability to support ad hoc category definition and retrieval in real-time.

In meeting these specifications, we were obligated to adhere to the conditions described in the following section.

## 3. Development of Automated Classification Techniques

Maintaining continuity with the existing RCDC system was a required condition for the integration of automated techniques into the grant categorization process. In terms of system requirements, this meant maintaining compatibility with the implemented vector space information retrieval model. Therefore, we restricted ourselves to linear classifiers, primarily linear support vector machines ([2], [3]), although experiments with linear perceptron show similar results ([4]).

Within the current RCDC application, the definitions of funding categories (i.e., the set of query terms and their ranking) are as important to subject matter experts as the set of retrieved documents. In practical terms, this means that not only should the output of a trained classifier be interpretable, it must be editable as well. Specifically, users requested the ability to:

1. limit the number of dimensions in the trained classification function;

2. delete dimensions that are intuitively irrelevant;

3. change the value of a dimension to match their intuitions about its relative importance;

4. add dimensions that were not part of the automatically generated classification function.

With regard to the documents retrieved or intended to be retrieved by a given query, users required the ability to indicate retrieval errors (false positives) and to augment the training data with externally labeled documents (false negatives) and incorporate these labels iteratively into the classifier's training procedure. In light of this requirement, we found that users should be given feedback about the inherent separability of the documents they are trying to classify. We found such feedback useful in tempering expectations of the performance of the automatic classifier and helping users understand why classification accuracy varies for different funding categories.

In the next section, we take the requirements above as our positions on the design of a document classification and retrieval tool that incorporates user interaction into the training procedure for an automatic classifier.

## 4. Positions on Designing an Interactive Automatic Document Classification System

In this section, we outline our positions on the design of an interactive document classification and retrieval tool that we prototyped for the RCDC project. Our positions are grounded in a series of experiments that measured classification accuracy for four categories – Lung Cancer, Breast Cancer, Prevention, and Orphan Drug. The data consisted of about 25,000 labeled grants that were funded by NIH in 2008 (a subset of about 80,000 total applications funded for the year).

### 4.1 Users must be able to edit the trained classification function.

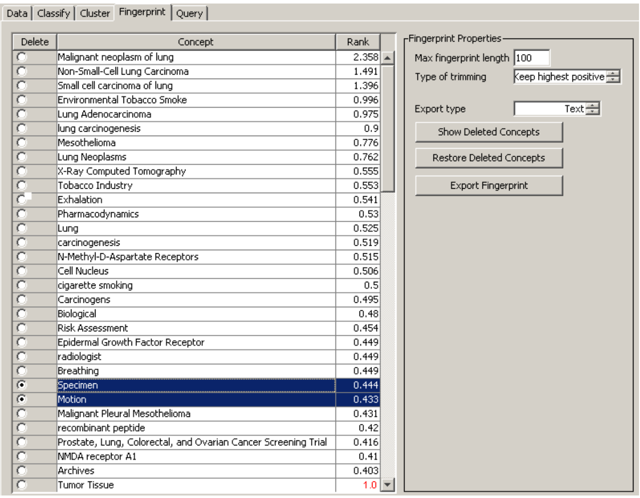
In an interactive document classification system, the trained classification function must be interpretable by users. By default, linear classifiers like perceptron or SVM produce as output a weighted list of all (or nearly all, in the case of SVM) of the input dimensions. In our case, this vector typically contained around 20,000 dimensions, which is too many for a user to make sense of. Therefore, the first step we took in producing a human-interpretable classification function was to limit the number of dimensions over which the classifier operated. We evaluated several techniques for restricting the dimensionality of the training data and found that a simple, two-pass strategy worked as good as anything:

1. Train the classifier on the full-dimensional data set.

2. Remove all but the top-*n* ranked dimensions in the output function from each item in the data set.

3. Retrain the classifier on the n-dimensional data set.

Happily, the optimal number of dimensions in our classification tasks turned out to be quite small (around 25-100 dimensions) relative to the dimensionality of the original data set. Another simple and effective way to restrict the dimensionality of the trained classification function is to remove low-ranked features from the data set before initially training the classifier (i.e., remove terms from a document vector if their weight is below some threshold).



| Delete | Concept | Rank |
|---|---|---|
| ○ | Malignant neoplasm of lung | 2.358 |
| ○ | Non-Small-Cell Lung Carcinoma | 1.491 |
| ○ | Small cell carcinoma of lung | 1.396 |
| ○ | Environmental Tobacco Smoke | 0.996 |
| ○ | Lung Adenocarcinoma | 0.975 |
| ○ | lung carcinogenesis | 0.9 |
| ○ | Mesothelioma | 0.776 |
| ○ | Lung Neoplasms | 0.762 |
| ○ | X-Ray Computed Tomography | 0.555 |
| ○ | Tobacco Industry | 0.553 |
| ○ | Exhalation | 0.541 |
| ○ | Pharmacodynamics | 0.53 |
| ○ | Lung | 0.525 |
| ○ | carcinogenesis | 0.519 |
| ○ | N-Methyl-D-Aspartate Receptors | 0.515 |
| ○ | Cell Nucleus | 0.506 |
| ○ | cigarette smoking | 0.5 |
| ○ | Carcinogens | 0.495 |
| ○ | Biological | 0.48 |
| ○ | Risk Assessment | 0.454 |
| ○ | Epidermal Growth Factor Receptor | 0.449 |
| ○ | radiologist | 0.449 |
| ○ | Breathing | 0.449 |
| ● | Specimen | 0.444 |
| ● | Motion | 0.433 |
| ○ | Malignant Pleural Mesothelioma | 0.431 |
| ○ | recombinant peptide | 0.42 |
| ○ | Prostate, Lung, Colorectal, and Ovarian Cancer Screening Trial | 0.416 |
| ○ | NMDA receptor A1 | 0.41 |
| ○ | Archives | 0.403 |
| ○ | Tumor Tissue | 1.0 |

Fingerprint Properties

Max fingerprint length: 100

Type of trimming: Keep highest positive

Export type: Text

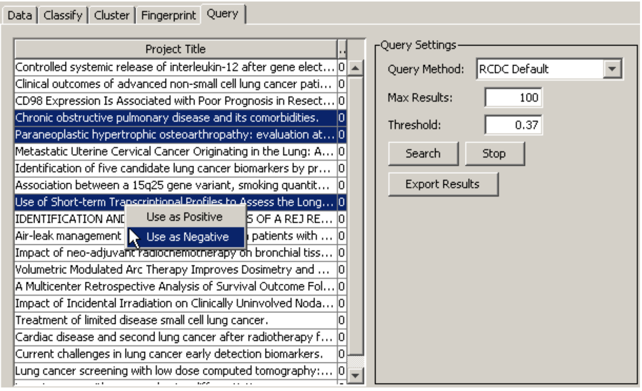
**Figure 1. Screenshot illustrating functionality to edit the trained classification function by deleting features or changing their weights.**

Even after limiting the number of features in the classification function, some items remain which are unacceptable to users' intuitions about whether they belong. For example, if a number of Lung Cancer grants originated from the N.C. Cancer Hospital, it is possible that a term like "North Carolina" would be heavily weighted in the resulting classification function. However, a user might feel that this term is inappropriate for a query intended to define Lung Cancer grants in general. Therefore, our prototype allows users to indicate that particular features should be excluded from the training data.

Figure 1 shows a screenshot of a prototype application that illustrates this functionality. The primary view in Figure 1 contains a set of weighted features which represent the decision boundary for a sample document category. In this illustration, the two blue highlighted terms have been selected by the user for removal from the classification function. The highlighted features will be removed from the training data prior to retraining the classifier, and in essence become inactive for any future data points that may contain them. We found that allowing users to selectively remove unwanted features from the training data generally has very little impact on classification accuracy and occasionally improves it.

Sometimes users want to override automatically computed weights for a given feature, usually because they feel that it merits a higher weight than the classifier assigned it. The problem with attempting to manually adjust feature weights is that they are likely to change with the next training cycle. Our solution to this requirement has been to store user-modified concepts separately, train the classifier, and overwrite those user-modified concepts before the classification stage. In Figure 1, the weight of the last term shown (“Tumor Tissue”) has been overridden by the user to 1.0 and is depicted in red font.

Another way to allow users to interactively modify classifier output is to allow them to apply the classifier to an unlabeled or partially labeled document set. The documents that are returned can be designated by the user as positive or negative examples and incorporated into retraining the classifier (Figure 2).



| Project Title | .. |
|---|---|
| Controlled systemic release of interleukin-12 after gene elect… | 0 |
| Clinical outcomes of advanced non-small cell lung cancer pati… | 0 |
| CD98 Expression Is Associated with Poor Prognosis in Resect… | 0 |
| Chronic obstructive pulmonary disease and its comorbidities. | 0 |
| Paraneoplastic hypertrophic osteoarthropathy: evaluation at… | 0 |
| Metastatic Uterine Cervical Cancer Originating in the Lung: A… | 0 |
| Identification of five candidate lung cancer biomarkers by pr… | 0 |
| Association between a 15q25 gene variant, smoking quantit… | 0 |
| Use of Short-term Transcriptional Profiles to Assess the Long… | 0 |
| IDENTIFICATION AND S OF A REJ RE… | 0 |
| Air-leak management patients with … | 0 |
| Impact of neo-adjuvant radiochemotherapy on bronchial tiss… | 0 |
| Volumetric Modulated Arc Therapy Improves Dosimetry and … | 0 |
| A Multicenter Retrospective Analysis of Survival Outcome Fol… | 0 |
| Impact of Incidental Irradiation on Clinically Uninvolved Noda… | 0 |
| Treatment of limited disease small cell lung cancer. | 0 |
| Cardiac disease and second lung cancer after radiotherapy f… | 0 |
| Current challenges in lung cancer early detection biomarkers. | 0 |
| Lung cancer screening with low dose computed tomography:… | 0 |

Query Settings

Query Method: RCDC Default

Max Results: 100

Threshold: 0.37

**Figure 2. Using document relevance feedback to relearn the decision boundary.**

## 4.2 Users must be given feedback on the inherent separability of their data set.

We sometimes observed users trying to distinguish document classes that are at best poorly separable over the given feature representation. For example, a category like Orphan Drug is defined axiomatically as research pertaining to any pharmaceutical agent used to treat a disease or condition that affects less than 200,000 people in the United States [5]. It is difficult to train a linear classifier to learn this distinction over document term vectors and produce an easily interpretable classification function. In cases like this, users tend to get disgusted at the classifier’s poor performance and blame the computer.

We found it useful to guide users’ expectations of system performance by visualizing document similarities with a two-dimensional interactive cluster plot (Figure 3). When users see high overlap between positive and negative training samples, they understand that they’re asking the classifier to do something hard. For example, in the top plot in Figure 3, there is relatively poor separation of the sample document class (green squares labeled “Positives”) from the remainder of the document set (red squares labeled “Negatives).

After the same data set has been restricted to 500 dimensions (from an original feature set of 7814 in this case), we observe less overlap of the two classes (bottom plot in Figure 3). The reduced feature set also corresponds to higher measures of classifier performance on a labeled test set. However, we found the type of visual feedback depicted in Figure 3 more effective than providing traditional evaluation measures like precision, recall, and F1 to users not accustomed to thinking in these terms.

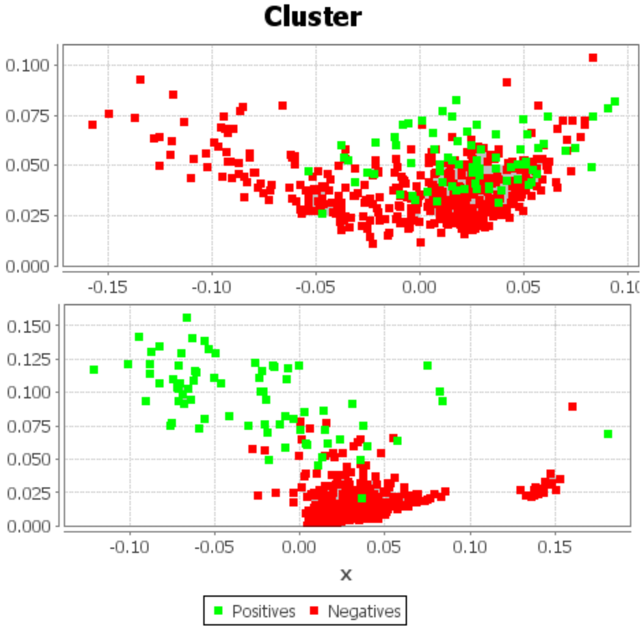
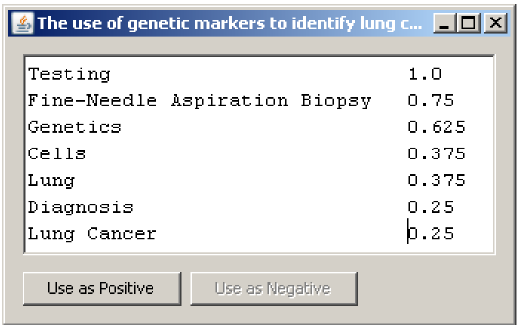


**Figure 3. Document separation and number of features. In the top panel distances between documents are computed over the full set of 7814 features before applying dimensionality reduction necessary for two-dimensional visualization. In the bottom panel only 500 features were retained.**

We also found it useful to allow users to interact with the classifier via the cluster plot (e.g., using the mouse to zoom or select individual data points). For example, after removing selected features from the training data, a user can examine the impact of this action by re-clustering the data and looking for relatively more or less class separation.

Alternatively, a user can select a document from the cluster plot to indicate whether it should be used as a positive or negative training point to retrain the classifier. Figure 4 shows a sample window that pops up when a data point is selected in the cluster plot. From here the user can see information such as the document title and weighted features that are active in the document, and provide relevance feedback to the classifier prior to any additional training iterations.



```
Testing                          1.0
Fine-Needle Aspiration Biopsy    0.75
Genetics                         0.625
Cells                            0.375
Lung                             0.375
Diagnosis                        0.25
Lung Cancer                      0.25
```

**Figure 4. Sample document popup window when a data point is selected in the cluster plot.**

## 5. Acknowledgments

We are indebted to Reinder Verlinde who provided an extremely useful wrapper to libsvm and essentially kick-started this whole prototype.

## REFERENCES

[1] Jablonski, S. 2008. Jablonski's Dictionary of Medical Acronyms and Abbreviations. 6th ed. Elsevier Health Sciences.

[2] C.-W Hsu, C.-C. Chang, and C.-J. Lin. (2003). A practical guide to support vector classification. Technical report, Taipei. http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf.

[3] Thorsten Joachims (2002). Learning to Classify Text using Support Vector Machines. Vol. 668 of Kluwer International Series in Engineering and Computer Science. Kluwer.

[4] Calvin Johnson, William Lau, Archna Bhandari, and Timothy Hays. (2008). A Best-Fit Model for Concept Vectors in Biomedical Research Grants. AMIA 2008 Symposium Proceedings: 93.

[5] RARE DISEASES ACT OF 2002. 107th Congress. http://frwebgate.access.gpo.gov/cgi-bin/getdoc.cgi?dbname=107_cong_public_laws&docid=f:publ280.107

# A Graphic User Interface for Content and Structure Queries in XML Retrieval

Luis M. de Campos, J.M. Fernández-Luna, Juan F. Huete, Carlos J. Martín-Dancausa
Departamento de Ciencias de la Computación e Inteligencia Artificial
E.T.S.I. Informática y de Telecomunicación. CITIC–UGR
Universidad de Granada, 18071 Granada, Spain
(lci,jmfluna,jhg,cmdanca)@decsai.ugr.es

## ABSTRACT

Structured Information Retrieval works with documents internally organised around a well defined structure, typically XML documents. In this research field, documents are not retrieved as a whole, but only those most specific relevant parts of the documents are delivered to the users. Lots of models have been developed to deal with the new dimension of the internal organization. From the point of view of the users, the document structure should be an added value in order to retrieve relevant material, because they are able to specify structural hints, in the form of the types of elements to be retrieved and as restrictions over some elements. There are several ways to query a system specifying content and structure queries (natural and artificial languages), but few of them rely on graphic user interfaces, supporting the users to create queries that fulfil more accurately their information needs. In this paper, we present a graphic user interface with the aim of formulating these types of queries, where the users only have to state what they wish to retrieve and structural restrictions about it.

## Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]; H.5.2 [**User Interfaces**]

## General Terms

Human Factors, Algorithms

## Keywords

XML, NEXI, Content and Structure queries, Graphic User Interface, Structured Information Retrieval

## 1. INTRODUCTION

The increasing importance of the internal structure of documents has caused the evolution of the Information Retrieval (IR) field to a new area where this important feature of the documents has been taken into account for retrieval purposes. The Structured IR field [1] deals with models and tools to index, retrieve and present *structured* documents, i.e. those which are internally organised around a well-defined structure. This means that classic IR models and techniques, which treats documents as if they where atomic, have been extended and adapted to exploit a more elaborate content representation, but also new ones have been specifically designed to tackle with this new challenge. By using SGML or XML mark-up metalanguages, structured documents can be easily represented and described.

Then the atomization of the flat documents is no longer considered, so the new view of such documents is an aggregation of interrelated units which need to be indexed, retrieved, and presented both as a whole and separately, in relation to the user's needs. In other words, an IR system must retrieve a set of document components (units) that are most relevant to this query, not just entire documents.

The inclusion of the structure of a document in the indexing and retrieval process affects the design and implementation of the IR system in many ways. First of all, the indexing process must consider the structure in an appropriate way so that users can search the collection both by content and structure. Secondly, the retrieval process should use both structure and content when estimating the relevance of documents. Finally, the interface and the whole interaction must enable the user to make full use of the document structure. The INEX (INitiative for the Evaluation of XML Retrieval) proceedings is an excellent source of information[1].

Focussing on the query formulation stage, there are several approaches for expressing an information need to a structured IR system: 1. The classical IR approach of providing a set of keywords, known as a *content-only (CO)* query in INEX terminology. 2. The use of a query language specifically designed for querying structured documents. 3. The use of a graphic user interface for formulating the query.

The main advantage of having a structured collection is that the structure could be exploited for retrieval purposes, as the user can specify in the query *what* she/he is looking for, and *where* this should be located in the required documents. The "what" involves the specification of the content, while the "where" is related to the structure of the documents. This kind of query is known as *content and structure (CAS) query* following the INEX convention. The output of the system would be a list of relevant XML elements (or elements of the required type) sorted by their relevance degree.

Then, with the first approach presented above, and from a point of view of a user interacting with a structured IR sys-

*HCIR* '09 Washington, DC USA

[1] See `http://inex.is.informatik.uni-duisburg.de/` and `http://www.inex.otago.ac.nz/` for INEX information.

tem, the structural restrictions in natural language are very difficult to be captured by the system, so the user is not getting the most of the document structure. In the second approach, a well defined query language allows the creation of CAS queries where structural restrictions are stated. The NEXI language [7] is the main representative of such languages, becoming a de facto standard in the textual facet of XML treatment. In this case, the user is able to declare what type of XML elements have to be returned by the system. The main problems of such languages are two: first, the user has to learn a relatively complex language, and later use it properly, and second, the structure of the collection has to be very well known by the user to take advantage of both the power of the query language and the collection itself. Finally, in the third option, by means of graphic components in a form, the system supports the formulation of a query. The two advantages are that a high knowledge of the structure of the collection is not required and a CAS query could be easily formulated. Also, the possibility of making a mistake in the formulation of the query is almost null because the process is controlled by the user interface. For expert users, the 2nd option perhaps is the most appropriate, but for common users, the 3rd option is the most suitable.

Some studies like [6] argue that the addition of users' structural requirements is not useful at all in XML-IR. One of the reasons is that the users are not able to formulate useful structured queries. Then more efforts have to be done in order to overcome this problem. In this line, this paper presents a user interface for formulating CAS queries in order to retrieve relevant elements from an XML collection. It is based on a Web form where the user is able to express the content query with its corresponding structural restrictions very easily and without a deep knowledge of the collection. This 'visual' query is translated to its corresponding NEXI query and passed to the retrieval system. This GUI has been implemented (and is being used) in *Seda*, an operational structured IR system for the official (structured) documents of the Andalusian Parliament[2].

After this introduction, in Section 2, we describe some related works and give basic notions of NEXI in order to understand the rest of the paper. Section 3 presents our proposal of user interface for CAS queries. Section 4 presents the algorithm designed to translate the visual query to the corresponding NEXI query. Finally, Section 5 contains our concluding remarks and various proposals for future research.

## 2. BACKGROUND AND RELATED WORKS

Although INEX has been a great success since its beginning in 2002 as evaluation forum and lots of new structured IR systems and models have been developed, always with a document-centric view of XML, and tested with the different collections adopted as official, few of them, even outside of this workshop, have focussed on the design of a user interface to facilitate the formulation of CAS queries.

Among them, and worried about the usability of the IR systems in terms of XML-related tasks, we could cite several works. Fuhr et al [4] build a user interface for XIRQL, a query language derived from XPath. They try to avoid to the user the learning of the syntax of XIRQL, and at the same time, to hide the structure of the collections. The main idea of this interface is having a specific area to formulate single query conditions, and a second one where the user can specify how the queries are combined. A third area is in charge of showing the XIRQL query.

In the same line as XIRQL, and in order to allow queries combining content and structure (CAS queries) to be specified, the NEXI language [7] was designed and considered with the time to XML-IR what SQL is to Databases. It is a simplified XPath containing only the descendant operator (//) in a tag path and also an extended XPath containing the *about* function. NEXI has been used by INEX since 2004, and it is usually the formal statement of a query, that will be passed to a structured IR system, to process it.

The kind of structured CAS query considered by NEXI can take two possible forms: a) $//C[D]$, which returns $C$ units about $D$, and b) $//A[B]//C[D]$, which returns $C$ descendants of $A$ where $A$ is about $B$ and $C$ is about $D$. $A$ and $C$ are *paths* (sequences of XML elements or structural units), specifying structural restrictions, whereas $B$ and $D$ are *filters*, which specify content restrictions, and // is the descendant operator. Each content restriction will include one or several *about* clauses, connected by either *and* or *or* operators; each about clause contains a text (a sequence of words or terms) together with a relative path, from the element which is the container of the clause to the element contained in it where this text should be located. $C$ is the *target* path (the last element in $C$ is the one that we want to retrieve) and path $A$ is the *context.*

An example of a NEXI-structured query is the following:

$$//A[about(., text2)\ and\ about(.//F, text3)\ and\ about(.//J, text4)]\\ //D[about(., text1)\ and\ about(.//N, text5)]$$

What we want to retrieve with this query are D elements which are contained within A elements. The target D elements should speak about *text1* and contain an N element speaking about *text5*; the context A elements should be about *text2* and also contain F and J elements dealing with *text3* and *text4*, respectively

NEXI is also able to represent CO queries with the wildcard '*': $//*[about(., text1]$. This expression means to retrieve any element relevant to text1.

Although NEXI is a relatively easy-to-use language, the formulation of queries with it usually requires a kind of expertise by the user, reason by which several techniques have been designed to avoid the direct use of NEXI, although the final query provided to the retrieval system is a NEXI query.

A first example is *NPLX* [8], which accepts natural language queries in a simple text field and generates NEXI queries. By an exhaustive analysis of the query by means of Natural Language Techniques (NPL), NPLX tries to find references to the document structure in the sentence and build an appropriate NEXI query.

A second example is *Bricks* [10], a query-by-template interface, that allows the user to input structured queries by means of text fields, for content needs, and list boxes, for structural needs. Later, when the query is formulated through the GUI, it is translated into NEXI and provided to the search engine. *Bricks* allows the user to formulate queries in several steps: first, the desired retrieval element, and later, additional information needs and restrictions.

A comparison of both alternatives in term of usability and effectiveness is presented in [9].

A totally different approach is the prototype *XmlBrowser* [5], where the users formulate queries by drawing a XML tree, where the nodes are the XML elements in the collection

[2] Andalusia is the southern region of Spain.

and the arcs are the structural relationships between the nodes. Constraints in nodes and arcs can be established, which basically are textual contents.

## 3. DESCRIPTION OF THE USER INTERFACE FOR CAS QUERIES

In this section, the user interface for supporting the formulation of CAS queries is presented. The final output of the query formulation process, and totally transparent to the user, will be a CAS query formulated in the NEXI language, which will be the input to any structured IR system able to process NEXI queries.

In order to reduce the complexity both in the query formulation by the user and in the NEXI generation process, the CAS queries that can be formulated in this user interface, present the following structure:

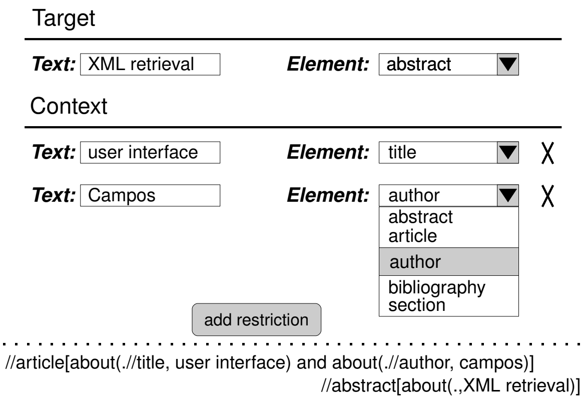
$$//A[about(.//B1, text1)\ and\ \ldots\ and\ about(.//Bn, textn)]$$
$$//C[about(., text0)\ and$$
$$about(.//Cm, textm)\ and\ \ldots\ and\ about(.//Cz, textz)],$$

i.e. only one type of retrievable element, C, with its associated textual query (and 0 or more abouts clauses – Cn,. . . ,Cz), plus 0 or more context clauses (abouts) (B1,. . . , Bn).

Therefore, and considering the components of this NEXI query, we have to design an appropriate visual method to express a target element (the document element in which the user is interested for retrieval purposes) and its associated text query, as well as the context element(s) (the document element(s) that establishes a restriction over the target elements) and its (their) corresponding text query(ies).

For this purpose, the form is composed of two groups of graphic components: those used to input the information related to the target, and those for the context. More specifically, in each group, the user will find a list box, where she/he could select a unit from, plus its associated text field, where the query terms are introduced in it. In both cases, for the target and context, the list boxes will contain comprehensible labels of the XML tags, instead of their names in the documents themselves, so they are totally transparent to the users. Specifically, for the target list box, only those retrievable tags are shown, while for the context, only those tags where restrictions can be established, are included. Leaving the text field blank and no element selected from the list box from the context group, the NEXI query will be only composed of the target part ($//C[about(., text0)$).

In Figure 1, we may see a design of the interface, according to the requirements given in the previous paragraphs. The two differentiated parts, the target and context groups, are represented. In the former, the text field and the list box are used to select the type of retrievable element and the textual query. In the case of the example of the figure, the user is pointing out that she/he is interested in abstracts dealing with “XML retrieval”, in the context of a collection of scientific articles. In the latter, following the same philosophy, the user is able to input the context of the search, i.e. restrictions imposed by the selection of other types of elements and the formulation of the associated query. In the example, the restrictions for those abstracts are that preferably contained in articles with titles about “user interface” and the author “Campos’. The lower part of the figure (not included in the interface) shows the NEXI query generated with the visual query. The user could include the number of restrictions that she/he consider (using the buttom with the text “add restrictions“) and, once formulated, she/he could remove any of them (clicking in the corresponding black cross on the right hand side). If the user selects the special label in the list box of the target group named ’any’, she/he is asking the structured IR system to return any type of relevant element.


Target

**Text:** XML retrieval **Element:** abstract

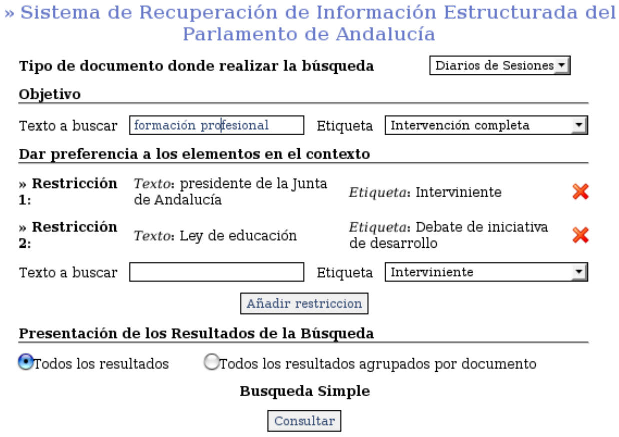
Context

**Text:** user interface **Element:** title

**Text:** Campos **Element:** author

//article[about(.//title, user interface) and about(.//author, campos)]
//abstract[about(.,XML retrieval)]

Figure 1: The user interface for CAS queries.


» Sistema de Recuperación de Información Estructurada del Parlamento de Andalucía

**Tipo de documento donde realizar la búsqueda** Diarios de Sesiones

**Objetivo**

Texto a buscar: formación profesional Etiqueta: Intervención completa

**Dar preferencia a los elementos en el contexto**

» **Restricción 1:** *Texto*: presidente de la Junta de Andalucía *Etiqueta*: Interviniente

» **Restricción 2:** *Texto*: Ley de educación *Etiqueta*: Debate de iniciativa de desarrollo

Texto a buscar Etiqueta: Interviniente

**Presentación de los Resultados de la Búsqueda**

● Todos los resultados ○ Todos los resultados agrupados por documento

**Busqueda Simple**

Figure 2: User interface for CAS queries in *Seda*.

As mentioned before, this design has been implemented in *Seda*, an operational structured IR system to retrieve official publications of the Andalusian Parliament [2], marked up in XML. The search engine underlying this system is *Garnata*, implementing a retrieval model for structured documents based on Bayesian networks and Influence Diagrams [3]. Figure 2 shows a screen shot of the visual query formulation interface (in Spanish)[3] with other additional features.

Once the basic components of the interface have been presented, it is the moment to stablish the differences with respect to *Bricks* [10], the most similar approach found in the specialized literature. The main one is that the user in this interface must select an element, from a list box, which will be the first in the path in the NEXI query, i.e. the root element after // (’In’ in their terminology). From that element, she/he must to select the retrievable element in which she/he is interested and its associated text query (’find’ and

[3] The user is requesting a complete speech dealing with “professional training“, where the speaker is the President of the Andalusian government, and integrated in a debate of a political initiative related to the “education law“.

'about' in their notation), and some restrictions, again selecting one or more pairs of element and associated query text ('with' and 'about' following their terminology). We think that if a user interface of this class has to assume an almost total lack of knowledge of the structure of the documents in the collection, to leave the decision of selecting the root element of the query to the user is not a good option. In our case, this decision is totally transparent to the user, only providing the target and restrictions, which is very intuitive.

A second difference, consequence of this design, is that the construction of the NEXI query in *Bricks* is direct, as they have the first element of the NEXI query (the 'A' element of the example query of Section 2), in contrast to our approach, because with the information provided by the user, that first element has to be determined. In the following section, the method designed to generate the NEXI query is presented.

## 4. FROM VISUAL QUERY TO NEXI QUERY

In order to convert a visual query to a NEXI query, the following data, extracted from the user interface, are required as input of the process: *target_element* (the desired type of elements to be retrieved), and the text query *target_text* for that target element, plus a set of pairs (*context_element1*, *context_text1*),...,(*context_elementN*,*context_textN*), contextualizing the target element, and finally the collection *Document Type Definition* (DTD). The output is a NEXI query with the following pattern:

$$//pivot\_element[context\_about\_list]$$
$$//target\_element[target\_about\_list].$$

Then, the translation process will have to find the different components of this NEXI query from the input data.

Once the XPath of all the different elements involved in the query are determined from the DTD, the first step is to find the *pivot_element.* This is performed extracting, from the set of paths composed of *target_element* and the $N$ *context_element*'s that contain the path of *target_element*, the common path of all of them. The last element in this path is considered the *pivot_element.*

With respect to *context_about_list*, it will be composed of $N$ about clauses joined by the 'and' operator. Inside each about, the element restriction is '.' if the paths from the *target_element* and the *context_elementi* are the same, or the last element in the *context_elementi* path, otherwise. The text of the about clause will be *context_texti.*

Finally, *target_about_list* is composed of several about clauses connected with the 'and' operator. The first about is related to the target element, containing a '.' in the element part and *target_text* in the text part. The rest of abouts come from those *context_element*'s whose paths contain the path of *target_element.* Specifically, the element part of the about clause is the last element of the path of *context_elementi.* The text part is its associated *context_texti.*

When "any" is selected from the available labels in the list box, *target_element* equals '*'. Finally, if no context is provided, then the NEXI query is $//target\_element[about(., target\_text)]$.

In general, this is an efficient method that mainly works with string operations. The generation of the NEXI query is very fast, negligible by the user.

## 5. CONCLUSIONS AND FURTHER WORK

This paper has presented a graphic user interface used to facilitate the formulation of CAS queries by the user, without the need of knowing any XML query language and being an expertise in the internal structure of the XML collection. It is composed of two main graphic component groups, one for specifying the target of the CAS query and other for indicating the context or restrictions. In both cases, the user select from a list of descriptive labels the XML elements in which she/he is interested and input the associated text queries. With these data, a NEXI query is constructed by mean of a simple procedure, and passed to the search engine in charge of the retrieval of the relevant elements. We think the presented interface is very intuitive and easy to use, facilitating the always complex process of giving expression to the user's information need.

With respect to the further research, as this interface is working on an operational system, and we know, from the users' feedback, that is easy and powerful, in spite of this, we are designing a usability study in order to know more formally the users' thoughts about it, as well as objective measures. The objective then is to improve it to overcome the possible problems. In addition, as we are designing XML relevance feedback techniques for the underlying search engine, *Garnata*, we are planning to re-design the user interface in order to incorporate this new feature.

### *Acknowledgements*

Work jointly supported by the Spanish Ministerio de Ciencia e Innovación (TIN2008-06566.C04.01) & research programme Consolider Ingenio 2010: MIPRCV (CSD2007-00018).

## 6. REFERENCES

[1] Chiaramella, Y. (2001). Information retrieval and structured documents. LNCS, 1980, 291–314.

[2] de Campos, L.M., Fernández-Luna, J.M., Huete, J.F., Martín-Dancausa, C., Tagua-Jiménez, A., Tur-Vigil, M.C. (2009). An integrated system for managing the Andalusian Parliament's digital library. Program: Elect. Lib. and Inf. Sys., 43(2), 156-174.

[3] de Campos, L.M., Fernández-Luna, J.M., Huete, J.F., Romero, A.E. (2007). Influence diagrams and structured retrieval: Garnata implementing the SID and CID models at INEX'06. LNCS., 4518, 165–177.

[4] Fuhr, N., Großjohann, K., Kriewel, S. (2003). A Query Language and User Interface for XML Information Retrieval. LNCS., 2818, 59–75.

[5] Harrathi, R., Calabretto, S. (2007). A query graph for visual querying structured documents. In Proc. Int. Conf. Digit. Inf. Manag., 116–120.

[6] Trotman, A. & Lalmas, M. (2006). Why structural hints in queries do not help XML-retrieval. SIGIR, 711–712.

[7] Trotman, A. & Sigurbjörnsson, B. (2005). Narrowed Extended XPath I (NEXI). LNCS., 3493, 16–40.

[8] Woodley, A., Geva, S. (2004). NPLX - An XML-IR system with a natural language interface. In Proc. 9th Australasian Doc. Comp. Symp., 71–74.

[9] Woodley, A., Geva, S., Edwards, S.L. (2006). Comparing XML-IR Query Formation Interfaces. Australian J. of Int. Inf. Proc. Sys., 9(2), 64–71.

[10] van Zwol, R., Baas, J., van Oostendorp, H., Wiering, F. (2006). Bricks: The Building Blocks to Tackle Query Formulation in Structured Document Retrieval. Lect. Notes Comput. Sc., 3936, 314–325.

# The HCI Browser 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

In this paper, we introduce the HCI Browser, a Mozilla Firefox extension designed to support studies of Web search behaviors. The HCI Browser presents tasks to the user, collects browser event data as the user searches for information, records answers that are found, and administers pre- and post-task questionnaires. The HCI Browser is a configurable tool that HCI and IR researchers can use to conduct studies and gather data about users' Web information seeking behaviors. It is especially well suited for "batch mode" laboratory studies in which multiple participants complete a study at the same time, but work independently. The HCI Browser is open-source software and is available for download at: http://ils.unc.edu/hcibrowser

## 1. INTRODUCTION

Laboratory studies of Web information seeking (including searching, browsing, managing, and refinding information) often involve presenting tasks to users and observing their behaviors as they use a web browser to look for information requested by the task. Researchers are interested in factors such as the web pages visited, links clicked, the amount of time spent on each page, and the use of the back button. In addition, before and after each task, researchers may wish to administer short questionnaires to gather additional data about the participants' experiences.

Researchers have built tools to help support studies of web search behaviors and have noted the challenges involved with capturing naturalistic user behaviors for web search [4]. Below, we summarize main approaches to capturing web browsing data and describe representative data collection tools that have been developed. We then outline the needs and motivations that led us to build our new tool, the HCI Browser.

### 1.1 Data Collection Approaches

There are many approaches to observing and collecting data about information seeking behaviors. Here, we focus on technologies to support automatically collecting data about browsing events such as page loads, link clicks, use of browser menus and buttons, etc. Four main approaches are: 1) HTTP proxies, 2) using an external program to monitor web browser events, 3) writing a custom web browser with built-in instrumentation, and 4) adding instrumentation code to an existing web browser. Due to space limitations, we only briefly summarize these approaches below. Keller, et al. [4] is a good resource for more detail.

***HTTP proxies*** – Proxies intercept HTTP requests and can log data about URL page requests. Proxies can also add tracking code to pages before they are sent on to the client browser. However, proxies are somewhat limited in the types of data they can collect because they do not have full access to the user interface events (i.e. menus, buttons, keypresses, etc.) in the web browser itself.

***External monitoring programs*** – Many operating systems and web browsers have "hooks" that allow external programs to monitor user interface and application specific events. WebTracker [7] and WebLogger [6] are two tools that use this approach to monitor an active web browser, observing and recording events such as link clicks and the use of menus and buttons. Time-stamped events are then written to a log file. URLTracer [3] uses a similar approach to write a simple log of all the URLs visited during a browsing session. Researchers have also used "spyware" and other event monitoring tools to capture web browsing events. A strength of this approach is that typically the monitoring tool can be installed without interfering with the user's normal browser configuration.

***Custom Web Browsers*** – Writing a custom web browser gives a researcher a large amount of control over what events are monitored and how they are logged. However, building a custom browser can require a large amount of programming expertise and time. Often, a custom browser is not built from scratch, but instead uses a component such as the Microsoft Web Browser Control (WBC). The Curious Browser [2] is an example of a custom-built web browser than has special instrumentation code to log user interaction events. In a previous project, we also built a custom web browser using the WBC [1].

One of the major downsides to developing a custom browser is that that the user interface is likely to be somewhat different than the full-featured, widely adopted browsers that are familiar to most users (i.e. different than Internet Explorer, Firefox, or Safari). Researchers may not have time to re-implement all the features of available in mainstream browsers. These differences may alter user behaviors in ways that are not well understood.

***Extensions to an existing browser*** – Adding instrumentation code to an existing browser is a powerful approach that combines advantages of the other methods. The idea is simple – instead of using an external monitoring program or writing a custom browser – add code to an existing browser. Most modern browsers have support for third-party plug-ins and extensions. For example, Keller et al. [4] implemented a "browser helper object (BHO)" that can be loaded every time Internet Explorer is run and can log browsing activity. The Lemur IR toolkit project recently introduced the Lemur Query Log Toolbar using this approach [5]. It is an open source browser plug-in tool that captures events such as page loads, tab switches, and searches issued to major search engines. Versions are available for both Firefox and Internet Explorer.

### 1.2 Motivations

The tools described above are all valuable research tools, but none filled all the needs we have for a study that is investigating how

*HCIR 2009*, October 23, 2009, Washington, DC, USA.

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, 3) provide support for administrative aspects of conducting a laboratory 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 developed a Mozilla Firefox extension called the HCI Browser. The HCI Browser is open-source code and we have utilized some concepts from the open-source Lemur Query Log Toolbar project [5]. 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 designed to: present experimental tasks to users, collect and log browser event data, manage the opening and closing of windows as needed, and to administer optional pre- and post-task questionnaires.

## 2.1 HCI Browser Overview

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 (Figure 4). The session number and participant id are used to label the data that is recorded for this session and can be any string that the experimenter wishes to use. The starting task number specifies what task to start with based on the order given in the task configuration file (configuration files are described in section 2.2). It is useful in case there is a need to re-start the program at a particular task number.

After clicking "OK", an introduction screen is displayed (Figure 5). The *intro.txt* configuration file is used to specify what text to display on this screen. This screen is useful to provide instructions and general information to the participant before beginning the tasks. Also, the experimenter can enter the information on the start-up screen before the participant arrives and then leave this introduction on the screen as the first thing the participant will see when they sit down at the computer.

When the participant clicks "OK" on the introduction screen, an optional set of pre-task questions can be presented (Figure 6). These questions can be changed using a configuration file and can be of three types: multiple choice, Likert-type/semantic differential, and open answer. If the configuration file is not present, then the pre-task question screen is skipped.

Next, the participant is taken to the main browser window which displays the first task in the toolbar area (Figure 7). This is a standard Firefox browser and the participant can search, browse, and navigate as usual. We decided to display the task and controls in the toolbar area at the top of the window so that web pages designed to fit standard screen resolutions would not require horizontal scrolling. The tradeoff is that vertical space is taken by the toolbar display and thus pages may require more vertical scrolling. In future versions, we may implement options for displaying the tasks in the toolbar, in a sidebar, or with no task presentation area, to allow the experimenter to decide which configuration best suits their needs.

In Figure 7, the user has navigated to a particular web site that has information requested by the task. There are two buttons in the toolbar: "Found an answer on this page", and "Done with answers for this task". The participant can use these buttons to submit answers they find and indicate when they are done with the task. The number of answers submitted are displayed in the lower left of the toolbar, along with an indication of the maximum number of answers they may submit (these are configurable).

When the participant finds an answer and clicks the "Found an answer" button, the toolbar changes as shown in Figure 8. The URL of the page is automatically entered into the "URL of answer" box, and the user can type in text of the answer in the "Answer text" box.

When the participant wishes to submit an answer as one of their "official" answers for this task, they can click the "Submit this answer" button. If they are to submit additional answers for this task, the controls revert back to show buttons for "Found an answer" and "Done with answers". When they are done with finding answers or have found the maximum number of answers for this task (configurable), then the system will automatically close all opened tabs and windows and display the (optional) post-task questions (not shown here, but the interface is similar to the pre-task questions).

As with the pre-task questions, the post-task questions are configurable by the experimenter and may be left out. When the participant has completed the post-task questions (or if they are left out), the program will then advance to the next task. When the last task is reached, a message is displayed letting the participant know that they have completed all the tasks.

## 2.2 HCI Browser Configuration Files

Every time the HCI Browser is loaded, four configuration files are read: a *introduction file* with text to show to users on an introduction screen, a *task file* (Figure 1) with the text of the tasks to present to the user, a *pre-task questions file* (Figure 2) with a set of questions to be asked prior to each task, and a *post-task questions file* (not shown) with a list of questions to be asked after each task. The pre- and post-task questions can be of three different types: 1) multiple choice, 2) Likert-type / semantic differential, and 3) free-text/open response. Note that in Figures 1 and 2, line numbers are shown for illustration, but they are not part of the actual files.

The task file (Figure 1) contains the text of the tasks to present to the users. Each task has two lines. The first line specifies the text of the task. The second line indicates the maximum number of answers that can be submitted for that task. In a future version of the HCI Browser, we plan to implement a minimum number of answers that will be specified on this line also.

The pretask.txt file is used to configure the pre-task questions. Three question types are supported:

- MultipleChoice – displays the question text followed by a vertical list of the choices
- LikertType – displays the question text followed by a horizontal list of the choices (note: this can be used for Likert-type and semantic differential scales)
- OpenAnswer – displays the question followed by a free-response text box

```
01  How tall is the U.S. capital building in
Washington, DC?
02  1
03   What  is  being  done  to  help  reduce
childhood obesity in the U.S.?
04  1
05  What are some reported causes of global
warming?
```

**Figure 1. Example Task Configuration File**

```
01  MultipleChoice
02  3
03  What is your favorite flavor of ice cream?
04  Vanilla
05  Chocolate
06  Strawberry
07  ---
08  LikertType
09  5
10  Ice cream is one of my favorite foods.
11  Strongly agree
12  Agree
13  Neutral
14  Disagree
15  Strongly disagree
16  ---
17  OpenAnswer
18  What  toppings  do  you  like  on  your  ice
cream?
19  ---
```

**Figure 2. Example Pre-Task Questions Configuration File**

To understand the format of the configuration file, we will step through the example in Figure 2. The first line of a question specifies the question type (lines 01, 08, and 17). For the MultipleChoice and LikertType questions, the next line specifies the number of answer choices. For example, lines 01 and 02 specify that this will be a multiple choice question with 3 answer choices. The next line (e.g. line 03) is the text of the question that will be displayed to the participant. Lines 04-06 specify the 3 answer choices for this question, one choice per line. The number of answer choices must correspond to the number specified on line 02. Finally, line 07 has exactly three dashes, and acts as a separator between questions. The LikertType question follows a format that is identical to the MultipleChoice described above. The OpenAnswer question only has two lines: one to specify the question type (e.g. line 17) and the next line to specify the text of the question (line 18).

## 2.3 Data Logging

In addition to collecting data from the questionnaires, while the performing the tasks (i.e. searching for the information), the HCI Browser monitors and logs of a wide array of browser events. The current version logs: pages loaded, links clicked, window and tab focus changes, open/close of windows and tabs, back/forward button clicks, URLs typed in the address bar, scrolling, history/bookmark menu activity. A new log file is automatically created for each task, and log entries include a timestamp, session number, participant number, and task number. An example section of a log file is shown in Figure 3.

The HCI Browser is available as open-source code. For information, downloads, and updates visit: http://ils.unc.edu/hcibrowser

# 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] Shaun Kaasten and Saul Greenberg. URL Tracer: URL Tracing Software for Internet Explorer. Retreived on August 24, 2009 from: http://grouplab.cpsc.ucalgary.ca/cookbook/ index.php/Utilities/URLTracer

[4] 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.

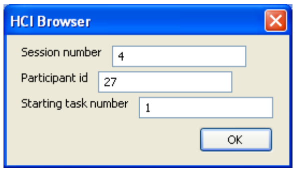
[5] Lemur Query Log Toolbar. http://www.lemurproject.org/querylogtoolbar

[6] 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.

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

```
1248198386334  21-7-2009   13:46:26 S4 P27   T1 intask   LoadCap  http://www.google.com/firefox?client=firefox-a&rls=or
1248198386398  21-7-2009   13:46:26 S4 P27   T1 intask   Focus    http://www.google.com/firefox?client=firefox-a&rls=or
1248198388384  21-7-2009   13:46:28 S4 P27   T1 intask   LClick   http://news.google.com/nwshp?client=firefox-a&rls=org
1248198389498  21-7-2009   13:46:29 S4 P27   T1 intask   Scroll   http://news.google.com/nwshp?client=firefox-a&rls=org
1248198389580  21-7-2009   13:46:29 S4 P27   T1 intask   Focus    http://news.google.com/nwshp?client=firefox-a&rls=org
1248198390167  21-7-2009   13:46:30 S4 P27   T1 intask   LoadCap  http://news.google.com/nwshp?client=firefox-a&rls=org
1248198401531  21-7-2009   13:46:41 S4 P27   T1 intask   Scroll   http://news.google.com/nwshp?client=firefox-a&rls=org
1248198411929  21-7-2009   13:46:51 S4 P27   T1 intask   RClick   http://news.google.com/news/url?sa=t&ct2=us%2F1_0_s_0
1248198413706  21-7-2009   13:46:53 S4 P27   T1 intask   AddTab   about:blank http://news.google.com/nwshp?client=firef
1248198413712  21-7-2009   13:46:53 S4 P27   T1 intask   Info     Currently open windows and tabs are logged below, but
1248198413718  21-7-2009   13:46:53 S4 P27   T1 intask   Info     Window=0, tab=0, url=http://news.google.com/nwshp?cli
1248198413724  21-7-2009   13:46:53 S4 P27   T1 intask   Info     Window=0, tab=1, url=about:blank http://news.google.c
1248198415971  21-7-2009   13:46:55 S4 P27   T1 intask   Scroll   http://news.google.com/nwshp?client=firefox-a&rls=org
1248198418755  21-7-2009   13:46:58 S4 P27   T1 intask   LoadCap  http://www.latimes.com/news/nationworld/nation/la-fg-
1248198419965  21-7-2009   13:46:59 S4 P27   T1 intask   SelTab   http://www.latimes.com/news/nationworld/nation/la-fg-
1248198420088  21-7-2009   13:47:00 S4 P27   T1 intask   Focus    http://www.latimes.com/news/nationworld/nation/la-fg-
1248198422564  21-7-2009   13:47:02 S4 P27   T1 intask   Scroll   http://www.latimes.com/news/nationworld/nation/la-fg-
1248198433480  21-7-2009   13:47:13 S4 P27   T1 intask   submittedAnswerURL   http://www.latimes.com/news/nationworld/n
1248198433488  21-7-2009   13:47:13 S4 P27   T1 intask   submittedAnswerText  The rare total eclipse will be visible th
```

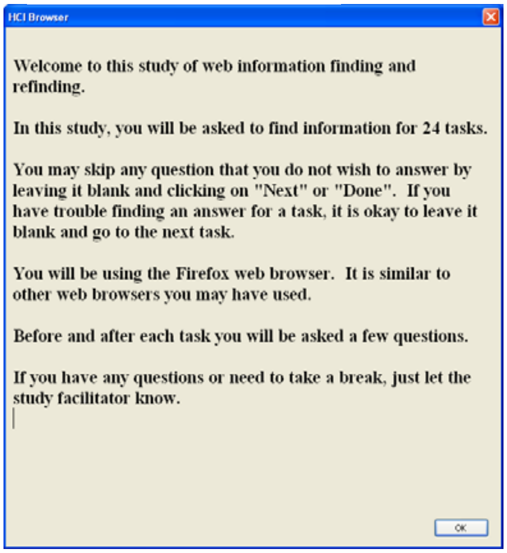
**Figure 3. Example Log File**



HCI Browser

Session number: 4
Participant id: 27
Starting task number: 1

**Figure 4. Start-up Screen**



HCI Browser

Welcome to this study of web information finding and refinding.

In this study, you will be asked to find information for 24 tasks.

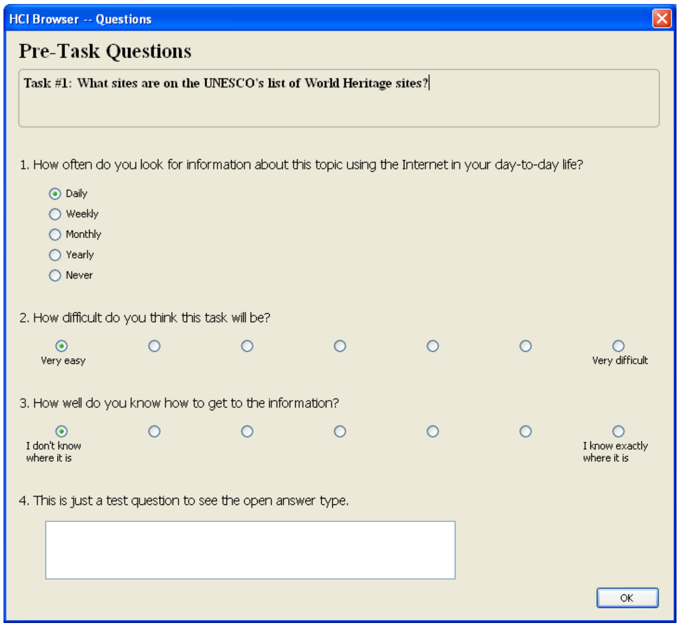
You may skip any question that you do not wish to answer by leaving it blank and clicking on "Next" or "Done". If you have trouble finding an answer for a task, it is okay to leave it blank and go to the next task.

You will be using the Firefox web browser. It is similar to other web browsers you may have used.

Before and after each task you will be asked a few questions.

If you have any questions or need to take a break, just let the study facilitator know.

**Figure 5. Introductory Text Screen**



HCI Browser -- Questions

Pre-Task Questions

Task #1: What sites are on the UNESCO's list of World Heritage sites?

1. How often do you look for information about this topic using the Internet in your day-to-day life?

● Daily
○ Weekly
○ Monthly
○ Yearly
○ Never

2. How difficult do you think this task will be?

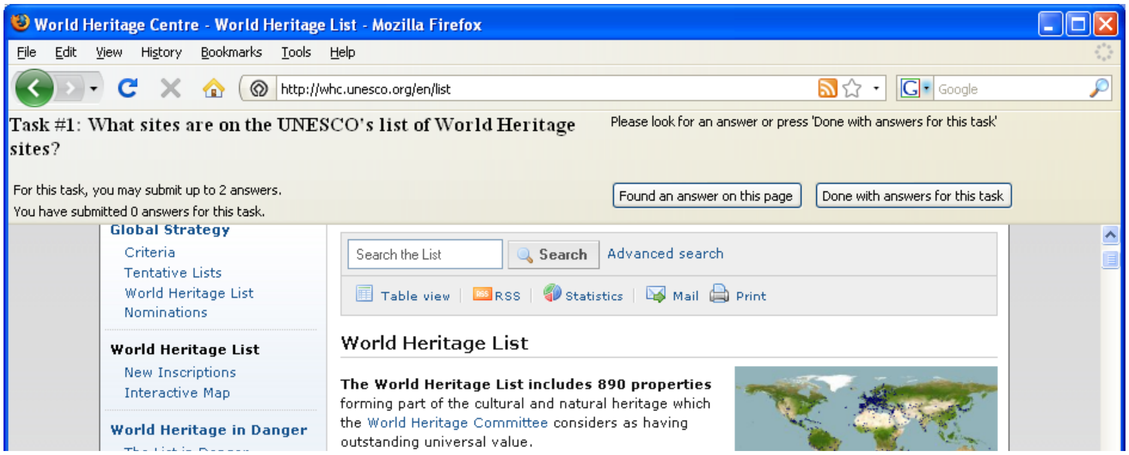
Very easy ● ○ ○ ○ ○ ○ ○ Very difficult

3. How well do you know how to get to the information?

I don't know where it is ● ○ ○ ○ ○ ○ ○ I know exactly where it is

4. This is just a test question to see the open answer type.

**Figure 6. Pre-Task Questions**



World Heritage Centre - World Heritage List - Mozilla Firefox

http://whc.unesco.org/en/list

Task #1: What sites are on the UNESCO's list of World Heritage sites?

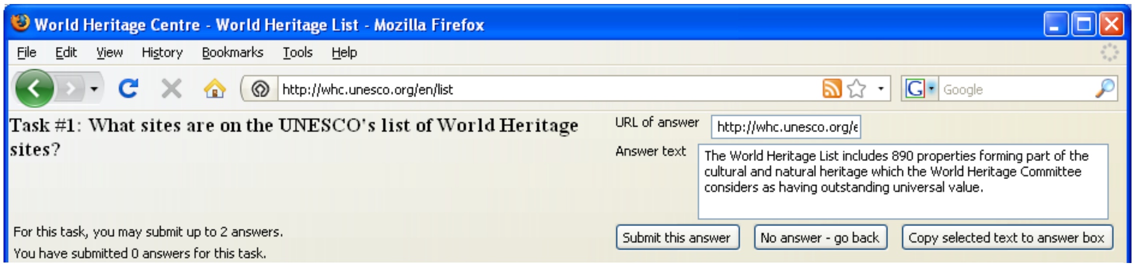
Please look for an answer or press 'Done with answers for this task'

For this task, you may submit up to 2 answers.
You have submitted 0 answers for this task.

World Heritage List

**The World Heritage List includes 890 properties** forming part of the cultural and natural heritage which the World Heritage Committee considers as having outstanding universal value.

**Figure 7. Task Presentation in Toolbar**



World Heritage Centre - World Heritage List - Mozilla Firefox

http://whc.unesco.org/en/list

Task #1: What sites are on the UNESCO's list of World Heritage sites?

URL of answer: http://whc.unesco.org/e

Answer text: The World Heritage List includes 890 properties forming part of the cultural and natural heritage which the World Heritage Committee considers as having outstanding universal value.

For this task, you may submit up to 2 answers.
You have submitted 0 answers for this task.

**Figure 8. Answer Submission**

# Improving Search-Driven Development with Collaborative Information Retrieval Techniques

Juan M. Fernández-Luna
Departamento de Ciencias de la Computación
e Inteligencia Artificial.
Universidad de Granada,18071.
Granada, Spain
jmfluna@decsai.ugr.es

Juan F. Huete
Departamento de Ciencias de la Computación
e Inteligencia Artificial.
Universidad de Granada,18071.
Granada, Spain
jhg@decsai.ugr.es

Ramiro Pérez-Vázquez
Departamento de Ciencias de la Computación.
Universidad Central de Las Villas, 50300.
Las Villas, Cuba
rperez@uclv.edu.cu

Julio C. Rodríguez-Cano
Departamento de Informática.
Universidad de Holguín, 80100.
Holguín, Cuba
jcrodriguez@facinf.uho.edu.cu

## ABSTRACT

Software developers frequently spend time searching for information, generally source-code. In the last few years this habit has increased the community's interest to improve it and some are staring to refer to as Search-Driven Development (SDD). In this work we examine the SDD as a collaborative and commonplace task. However, current integrated development environments (IDEs) do not include information retrieval systems with support for explicit collaboration among developers with shared technical information need. We then introduce PosseSrc, a prototype outside the IDEs that enables teams of remote developers to collaborate in real time during the search sessions. PosseSrc improve the SDD by supporting several modern state-of-the-art collaborative information retrieval (CIR) techniques such as session persistence, division of labor, sharing of knowledge and group awareness.

### Categories and Subject Descriptors

H.5.3 [**Information Interfaces and presentation (e.g., HCI)**]: Group and Organization Interfaces; H.3.3 [**Information Storage and Retrieval**]: Search Process.

### General Terms

Design, Human Factors

### Keywords

Search-Driven Development, Collaborative Information Retrieval, Collaborative Search, Source-Code Search, Multi-User Search Interface.

*HCIR* '09 Washington DC, USA

## 1. INTRODUCTION

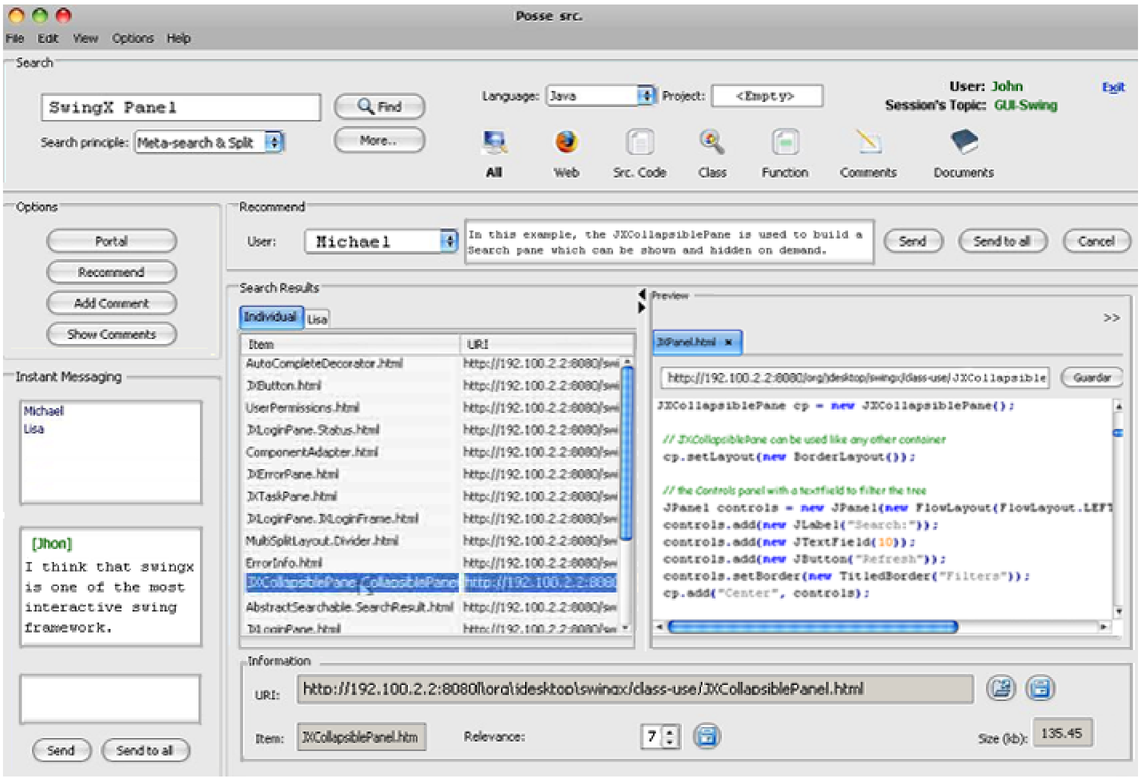
*"Good programmers know what to write. Great ones know what to rewrite (and reuse)".*
– E. S. Reymond[10]

You can rewrite or reuse good source-code, but first you must find it. That is the fundamental key by which some information retrieval (IR) systems have become a critical tools for software developers. Currently there are some specialized IR systems for source-code search. Examples include Google Code Search, Krugle, CodeFetch, Koders, and Codase. Some of then such as Koders can be integrated with IDEs such as Eclipse and Visual Studio.NET.

More recently there has been some significant efforts both from academia and the industry to fix SDD as a new research area motivated by the observation that software developers spend most of their times in searching pertinent information that they need to solve their task at hand [1]. K. Krugler and J. D. Mitchell remark in [6] that *"about 25% of a developer's time is spent searching for information. It's well spent, though – finding reusable code can get a project done on time and with high quality results"*.

In addition, software development can be considerated as a collaborative activity in which business analysts, customers, system engineers, architects, and developers interact among them. The concurrent edition of models and processes requires synchronous collaboration between architects and developers who cannot be physically present at a common location [5].

However, current IR systems do not have support for explicit collaboration among developers with shared technical information needs, which frequently look for additional documentation on the API, read newsgroups for people having the same problem, search the company's site for help with the API, and search for source code examples where other people successfully used the API [6]. Fortunately, in the last few years, some researchers have realized that collaboration is an important feature, which should be analyzed in detail in order to be integrated with professional IR systems, upgrading them to collaborative information retrieval(CIR) systems.



SwingX Panel

Search principle: Meta-search & Split

Language: Java

Project: <Empty>

User: John

Session's Topic: GUI-Swing

User: Michael

In this example, the JXCollapsiblePane is used to build a Search pane which can be shown and hidden on demand.

| Item | URI |
|---|---|
| AutoCompleteDecorator.html | http://192.100.2.2:8080/swi |
| JXButton.html | http://192.100.2.2:8080/swi |
| UserPermissions.html | http://192.100.2.2:8080/swi |
| JXLoginPane.Status.html | http://192.100.2.2:8080/swi |
| ComponentAdapter.html | http://192.100.2.2:8080/swi |
| JXErrorPane.html | http://192.100.2.2:8080/swi |
| JXTaskPane.html | http://192.100.2.2:8080/swi |
| JXLoginPane.JXLoginFrame.html | http://192.100.2.2:8080/swi |
| MultiSplitLayout.Divider.html | http://192.100.2.2:8080/swi |
| ErrorInfo.html | http://192.100.2.2:8080/swi |
| JXCollapsiblePane.CollapsiblePane | http://192.100.2.2:8080 |
| AbstractSearchable.SearchResult.html | http://192.100.2.2:8080/swi |
| JXLoginPane.html | http://192.100.2.2:8080/swi |

http://192.100.2.2:8080/org/jdesktop/swingx/class-use/JXCollapsible

```
JXCollapsiblePane cp = new JXCollapsiblePane();

// JXCollapsiblePane can be used like any other container
cp.setLayout(new BorderLayout());

// the Controls panel with a textfield to filter the tree
JPanel controls = new JPanel(new FlowLayout(FlowLayout.LEFT
controls.add(new JLabel("Search:"));
controls.add(new JTextField(10));
controls.add(new JButton("Refresh"));
controls.setBorder(new TitledBorder("Filters"));
cp.add("Center", controls);
```

Michael
Lisa

[Jhon]
I think that swingx is one of the most interactive swing framework.

URI: http://192.100.2.2:8080/lora/idesktop/swingx/class-use/JXCollapsiblePanel.html

Item: JXCollapsiblePanel.htm

Relevance: 7

Size (kb): 135.45

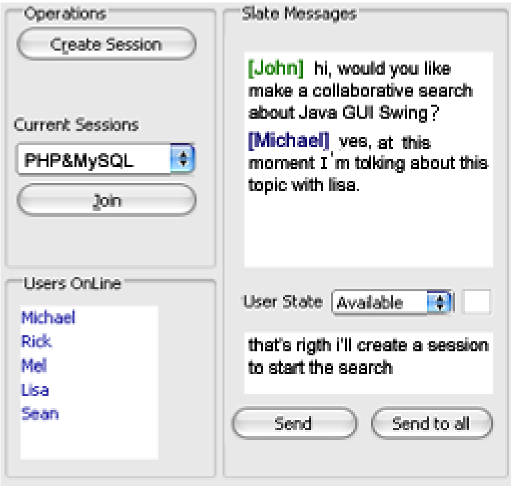
**Figure 1: PosseSrc prototype. Search: search control panel; options: main options buttons; recommend: recommender; search result: individual search result and recommendations; instant messaging: chat tool embedded; information: general item information; previewer: item selected viewer**

CIR is an emerging research field that belongs to a specialized area within the IR discipline. Therefore CIR includes the research areas that traditionally have been part of IR, but with an especial emphasis on the explicit collaboration among people with shared information needs. This fact requires that IR mixture with other disciplines such as Human-Computer Interaction (HCI) and Computer-Supported Cooperative Work (CSCW).

In this paper we introduce PosseSrc, a prototype outside the IDEs that enables remote team developers to collaborate in real time during the source-code search and any other related technical information. The design of PosseSrc was motivate by a brief survey that we applied to 50 students and professors related to software development projects in higher education in the domain of Information Technologies. Our survey results indicate that collaborative source-code search is a commonplace task. When we asked: *Have you ever collaborated with other programmers to search source-code?*, 78.0% responded yes. In addition, we asked: *Which are the activities that have motivated you to collaborate during the source-code search?*, the most common answers were: a) *meetings of the team members to clarify programming doubts while someone searches for source-code examples*, b) *dividing the search by each team member and sharing the final results*, c) *saving and documenting the search results of each one for sharing them*, and d) *consulting or answering doubts via chat or email.*

Based on survey respondents' descriptions, we identified four modern state-of-the-art CIR techniques for supporting collaborative source-code search. Session Persistence: storing a search session in a persistent format is a key requirement to facilitate collaboration during the session, revising the search at a later time, or sharing the results of a search with others [7]. Division of labor: Morris's survey in [7] describes ad-hoc methods to avoid duplication of effort during a searching task, such as dividing up the space of potential keywords, searching engines, or sub-tasks among different group members. Supporting mechanisms for dividing up and sharing work among participants is important for the success of a UI for multi-user search. Sharing of knowledge: in any collaborative setting there will be a large and diverse knowledge base shared among groups of members. Each one will bring their own experiences, expertise and topic knowledge to a particular searching task. What is needed is a way to enable the sharing of knowledge within the group [4]. Finally Group awareness: awareness is an essential element in distributed collaborative environments. Over the last decade, a number of researchers have explored the role of group awareness for supporting collaboration between distributed groups. Specifically in CIR, awareness is another key requirement [7].

We have organized this paper as follows. In the next section (2), we shall describe the main ideas of the design considerations of PosseSrc, its implementation and an overview of some related work with both CIR and SDD research areas. Finally, we conclude this paper in Section 3 by summarizing the exposed topics and our research future direction.



Michael
Rick
Mel
Lisa
Sean

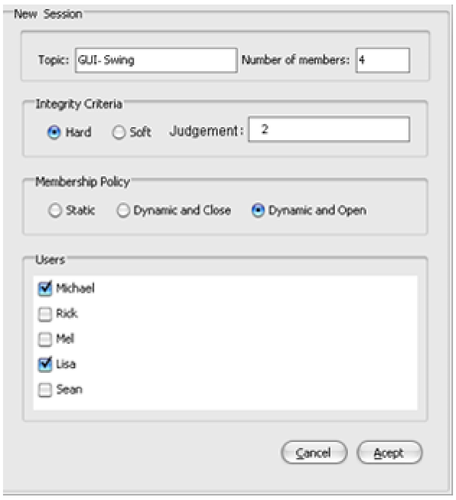
[John] hi, would you like make a collaborative search about Java GUI Swing?

[Michael] yes, at this moment I'm tolking about this topic with lisa.

User State: Available

that's rigth i'll create a session to start the search

**Figure 2: Collaborative portal**



Topic: GUI- Swing

Number of members: 4

● Hard ○ Soft Judgement: 2

○ Static ○ Dynamic and Close ● Dynamic and Open

☑ Michael
☐ Rick
☐ Mel
☑ Lisa
☐ Sean

**Figure 3: Creating a collaborative search session.**

## 2. POSSESRC

PosseSrc is designed to enable either synchronous or asynchronous, but explicit remote collaboration among team developers with shared technical information. Figure 1 shows the interface of the prototype client interface. The search-box wraps the search control panel (SCP), it permits to specify the developers queries, programming language or project on which the search will be accomplished. Moreover it can specify a searching field: comments, source-code, class or methods declaration, and whole source files. Rather interestingly, source-code search all by itself doesn't solve the whole problem. We need all of the technical information around and about the source-code to be able to really fly. For instance, the best examples of how to use some piece of source-code is often embedded as a small source-code snippet inside a magazine article or blog entry, or occasionally, even in the official technical documentation [6]. For these reasons in the SCP you can select documents or the Web as collections too.

The SCP also offers the possibility to specify the division of labor principle. It determines which principle to use to divide the search results among team developers: a)Meta-search engines and split: the results of each search engine available are merged in one results list, which is automatic divided among developers; b) Multi-search engines and switch: the results of each search engine available are switched among developers, where one developer can review only the results of an specific search engine; and c) Single-search engine and split: the results of the selected search engine are automatic divided among team developers. The options-box wraps the principal options of the PosseSrc that permits dynamic management of the GUI. For example, a developer can show the chat tool embedded (instant messaging-box), a collaborative portal where the developers can negotiate the creation of a collaborative search session (CSS) (Figure 2), a recommendations panel (recommend-box) to carry out explicit recommendations among developers, add and show comments of the current and historical search results, and the previewer panel (previewer-box) to review the results. In the search result-box the individual results and the recommendations made by others end-users are shown. The green-box permits getting specific information from the document selected in the results panel.

From the collaborative portal one developer can create a CSS, which consists of a team of developers working together to satisfy their shared technical information needs. For each CSS is necessary to establish (Figure 3). First, the main topic refereed to the shared technical information need of the developers. Second, the maximum number of developers allowed in the CSS. Third, the integrity criteria. The validity condition for minimum and maximum number of developers in a CSS. It can be: a) hard – the CSS is released if the integrity criteria are not satisfied; and b) soft – the CSS is suspended if the integrity criteria are not satisfied. And fourth, Membership Policy. It establishes how a potential member joins and leaves a session. All potential members can negotiate an invitation to join a session throughout the collaborative portal in different ways: a) static – a potential member must join to a CSS by previous negotiation and before the work has been started; b) dynamic and closed – each potential member must by explicitly invited to join the CSS; and c) dynamic and open – potential members can join a CSS on invitation or by own initiative at any time.

### 2.1 Implementation

For the implementation of PosseSrc we use CIRLab (Collaborative Information Retrieval Laboratory), a groupware framework for CIR research and experimentation [3], Java as programming language and AMENITIES (A MEthodology for aNalysis and desIgn of cooperaTIve systEmS) as software engineering methodology. CIRLab has been designed applying design patterns and an object-oriented middleware platform to maximize its reusability and adaptability in new contexts with a minimum of programming efforts. The distribution and communication facilities of CIRLab are ICE[1] (Internet Communications Engine) conforming. ICE applications are suitable for using them in heterogeneous environments: client and server can be written in different programming languages, run on different operating systems and hardware architectures, and communicate using a variety of

[1] http://www.zeroc.com

networking technologies. CIRLab also wraps some open-source three party APIs (e.g. search engines and a database engine). To do searches in different parts of the source-code (e.g. comments, class and function definitions) we extend CIRlab with parsers that allow indexing fields (parts of the source-code) when combined with search engines (e.g. Apache Lucene).

### 2.2 Related Work

PosseSrc include several areas of research, highlights of which CIR and SDD. On the one hand, some researchers have identified different search scenarios where is necessary to extend the IR systems with collaborative capabilities. For example, in the Web context, SearchTogether [8] is a system which enables remote users to synchronously or asynchronously collaborate when searching the Web. It supports collaboration with several mechanisms of group awareness, division of labor, and persistence. On the other hand, the SDD community present different prototypes and systems. For example, Sourcerer [2] is an infrastructure for large-scale indexing and analysis of open source code. Sourcerer crawls Internet looking for Java source-code from a variety of locations, such as open source repositories, public web sites, and version control systems.

In contrast to these approaches, PosseSrc makes a contribution in current SDD providing explicit support for teams of developers, enabling developers to collaborate on both the process and results of a search. It provides collaborative search functions for exploring and managing source-code repositories and documents about technical information in software development context. In order to support such CIR techniques PosseSrc provides some collaborative services. The embedded chat tool enables direct communication among different developers. Also relevant search results can be shared with the explicit recommender mechanisms. Another important feature enabling improvement is the automatic division of labor. Through awareness mechanisms all developers are always informed about the team activities to avoid the unnecessary duplication of effort. Awareness is a valuable learning mechanism that help the less experienced developers to view the syntax used by their teammates, and then be inspired to reformulate their queries. All search results can be annotated, either for personal use, like a summary, or in the team context, for discussion threads and ratings.

## 3. CONCLUSION AND FUTURE WORK

Novel CIR techniques such as session persistence, division of labor, knowledge sharing and awareness can be applied in several domains. For example, in the Web context, interactive multimedia, education and medical environment. We identified SDD as another applicable field, given both the collaborative nature and the interest in having specialized source-code search tools in the area of software development. In this sense we present PosseSrc, a prototype designed to enable either synchronous or asynchronous, but explicit remote collaboration among team developers with shared technical information need.

To conduct the PosseSrc's evaluation in a close future we identify the metric proposed by Pickens et al. in [9] as a good intention, where they proposed viewed precision ($P_v$, the fraction of documents seen by the user that were relevant) and selected precision ($P_s$, the fraction of documents judged relevant by the user that were marked relevant in the ground truth), and selected recall/viewed recall ($Rs/Rv$) as their dependent measures. Moreover, and taking into consideration our survey results, when 92.9% of our respondents use their workstation as an important dynamic collection of relevant information, we will add to PosseSrc on the base of CIRLab the capability of indexing local collections.

## 4. ACKNOWLEDGEMENTS

This work has been jointly supported by the Spanish Ministerio de Ciencia e Innovación (TIN2008-06566.C04.01) and Spanish research programme Consolider Ingenio 2010: MIPRCV (CSD2007-00018).

## 5. REFERENCES

[1] S. Bajracharya, A. Kuhn, and Y. Ye. Suite 2009: First international workshop on search-driven development - users, infrastructure, tools and evaluation. *International Conference on Software Engineering Companion*, 0:445–446, 2009.

[2] S. Bajracharya, J. Ossher, and C. Lopes. Sourcerer: An internet-scale software repository. In *SUITE '09: Proceedings of the 2009 ICSE Workshop on Search-Driven Development-Users, Infrastructure, Tools and Evaluation*, pages 1–4, Washington, DC, USA, 2009. IEEE Computer Society.

[3] S. Cleger-Tamayo, J. M. Fernandez-Luna, J. F. Huete, R. Perez-Vazquez, and J. C. Rodriguez-Cano. A proposal for an experimental platform on collaborative information retrieval. *International Symposium on Collaborative Technologies and Systems*, 0:485–493, 2009.

[4] C. Foley, A. F. Smeaton, and H. Lee. Synchronous collaborative information retrieval with relevance feedback. In *CollaborateCom 2006 - 2nd International Conference on Collaborative Computing: Networking, Applications and Worksharing*, pages 1–4, 2006.

[5] M. Jiménez, M. Piattini, and A. Vizcaíno. Challenges and improvements in distributed software development: A systematic review. 2009.

[6] K. Krugler and J. D. Mitchell. Search-driven development: Five reasons why search is your most powerful tool, 2007.

[7] M. R. Morris. A survey of collaborative web search practices. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 1657–1660, New York, NY, USA, 2008. ACM.

[8] M. R. Morris and E. Horvitz. Searchtogether: an interface for collaborative web search. In *UIST '07: Proceedings of the 20th annual ACM symposium on User interface software and technology*, pages 3–12, New York, NY, USA, 2007. ACM.

[9] J. Pickens, G. Golovchinsky, C. Shah, P. Qvarfordt, and M. Back. Algorithmic mediation for collaborative exploratory search. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 315–322, New York, NY, USA, 2008. ACM.

[10] E. S. Raymond. *The Cathedral and the Bazaar.* O'Reilly & Associates, Inc., Sebastopol, CA, USA, 1999.

# A visualization interface for interactive search refinement

Fernando Figueira Filho[*]
Institute of Computing
State University of Campinas
(Unicamp), Brazil
fernando@las.ic.unicamp.br

João Porto de Albuquerque
School of Arts, Sciences and Humanities
University of Sao Paulo, Brazil
joao.porto@usp.br

André Resende
Institute of Computing
State University of Campinas
(Unicamp), Brazil
resende@las.ic.unicamp.br

Paulo Lício de Geus
Institute of Computing
State University of Campinas
(Unicamp), Brazil
paulo@las.ic.unicamp.br

Gary M. Olson
Bren School of Information and Computer Sciences
University of California, Irvine, USA
gary.olson@uci.edu

## ABSTRACT

It is common practice nowadays to find, assess and explore the Web by groping scattered information presented through many search results. Browsing interfaces and query suggestion techniques attempt to guide the user by providing term recommendations and query phrases. In this paper, we introduce the browsing interface of Kolline, a community search engine under development. Two case studies are described and two distinct web browsing interfaces are analyzed. Based on this analysis, we present a new browsing interface, describing our design decisions and providing directions for future work.

## Categories and Subject Descriptors

H.5.3 [**Information Interfaces and Presentation**]: Group and Organization Interfaces—*web-based interaction, collaborative computing, organizational design*

## General Terms

Design, Human factors

## Keywords

Web 2.0, user-generated annotations, browsing interfaces

## 1. INTRODUCTION

In recent years, "Web 2.0" [3] applications have been employing tagging as a way for annotating published content. The greatest advantage of tagging systems is that they provide a means to gather the community vocabulary for further classification. Another important characteristic is that they carry different levels of specificity, ranging from very general, widely-used terms to domain-specific terms. This is especially useful in the case of on-line communities within which people are trying to interact and find shared content. In these environments, people may have different backgrounds and distinct areas of expertise, which leads to different perspectives on classification [1].

Regarding the user interface, there are two ways to take advantage of user-generated annotations when looking for information. First, the keyword-based search, which consists of a text box and a search button. The problem with this approach is that it assumes that the user knows how to formulate the query. This is especially hard for people who are trying to find information in different knowledge domains. A recently published article [5] points out that it is a common practice for researchers to find, assess, and exploit a range of information by scanning portions of many articles, instead of looking for a single article to read, in what the authors call "strategic reading". We also have seen this behavior within the open-source community. In order to solve a technical problem, sometimes one need portions of information that may be scattered throughout a series of different postings within a web forum. In these cases, finding the correct keywords which will lead to relevant results can be a time-consuming task. Term suggestion techniques attempt to address this issue, but still depend on an initial query in order to provide further suggestions. This issue is particularly relevant when exploring information in knowledge domains within which the user does not have a strong background, e.g. novice users searching for problem solutions in a web forum.

Second, many websites provide tag clouds or weighted lists, which consist of a visual depiction of user-generated annotations. In this approach, the criteria to show a given term is its use frequency, i.e. how many times users applied that term to annotate content. However, there are some problems with this type of visualization. On one hand, usually only popular terms are depicted, which might not be useful to a user who is searching within one or more specific

[*] Work done while at the University of California, Irvine.

*HCIR* '09 Washington, DC USA

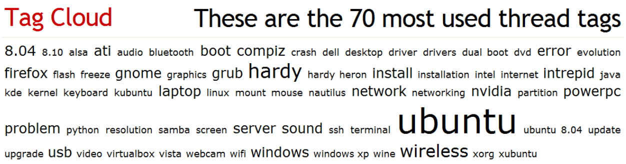
topics [8]. On the other hand, a user can choose only one term at a time and very often one need a conjunction of terms to suitably express the search task. Consequently, it is impossible to refine a search by only using the tag cloud.

With the aim of addressing those issues, this work presents the navigation interface of Kolline, a community search engine currently under development. It features a term recommendation tool, which suggests terms based on the user's previous interactions. The tool does not require that the user provides an initial query and the interactions can be done solely by clicking on the recommendation tool. In addition, users can refine their search context by choosing new terms which are semantically-related using an underlying ontology. The tool recommends terms which hold subsumption relationships, so a user can refine the search by clicking on general terms at first, and then narrow down the search context by choosing more specific terms. A text box is also provided, so the user can add, remove or modify terms during the interaction.

The paper is organized as follows: Section 2 describes the two cases we studied before designing our application. Section 3 explores some concepts and examples of browsing interfaces for searching, finally introducing our solution. The paper finishes with Section 4 providing an overview of our evaluation plan for the future and drawing conclusions.

## 2. CASE STUDIES

To characterize the problem, we explored two cases of visualization interfaces for user-generated annotations. First, the case of Ubuntu Forums[1], which represents the major source of information in the Web about this Linux distribution and unites an open-source community of developers and users interested in sharing information about troubleshooting, new features, and other related content. Users exchange information by adding new posts that are shown in the form of threaded discussions. Some users annotate these threads with tags, which helps to categorize content by using a community-oriented, non-controlled vocabulary. Fig. 1 shows a typical tag cloud containing the most frequent terms associated with threaded discussions in the forum.



Tag Cloud These are the 70 most used thread tags

8.04 8.10 alsa ati audio bluetooth boot compiz crash dell desktop driver drivers dual boot dvd error evolution firefox flash freeze gnome graphics grub hardy hardy heron install installation intel internet intrepid java kde kernel keyboard kubuntu laptop linux mount mouse nautilus network networking nvidia partition powerpc problem python resolution samba screen server sound ssh terminal ubuntu ubuntu 8.04 update upgrade usb video virtualbox vista webcam wifi windows windows xp wine wireless xorg xubuntu

**Figure 1: Ubuntu Forums' tag cloud**

To better understand whether this type of visualization agregates any significant value to finding information, we refer to a user study which attempted to assess the usefulness of tag clouds in comparison to the traditional keyword-based search. [8] conducted an experiment, giving participants the option of using both approaches to answer various questions. They found that while some participants preferred to use the text search box exclusively, a significant proportion of participants used the tag cloud to find information. There were two scenarios in which the tag cloud's use outweighed the search box's use: (a) when the information-seeking task was broad and non-specific, such as "paste the title of an article you find amusing or interesting" and (b) when the tag cloud contained a term relevant to the question. The first case obviously does not apply to answering technical questions. Most people who visit a Linux distribution forum are looking for help from other community members on a specific topic. However, it could be scenario (b), i.e. the user finds a term in the tag cloud which is relevant to answering a technical problem. But, since the cloud shows only the most used tags, it is not clear if this is sufficient to fulfill a specific search task. In other words, the users will probably have to refine their search by adding other terms. This refinement phase is important for reducing the overall number of hits and excluding irrelevant information from search results. However, most tag clouds or weighted lists do not provide this functionality.

Our second case is a community of professors and researchers of the University of São Paulo. The School of Arts, Sciences and Humanities is an interdisciplinary institute where professors hold positions in a great variety of research areas. To stimulate scientific collaboration, an institutional website is under development, which will contain information about each researcher, organized by area of expertise. In a first phase, professors were asked to provide terms which would describe their research interests and current activities. The union of all colected terms is shown as a list, but because of the great diversity of topics, the result does not fit in one page.

The problems with this visualization approach are twofold. First, each professor uses a particular level of specificity to describe his/her research area. General terms such as "molecular biology" are separated from specific terms like "proteins", although both research areas may have a certain level of intersection. A weighted list approach which shows the most frequent terms in order to reduce the list size is not a suitable solution, because it would not show specific terms that are relevant to the researchers. Second, professors working in the same areas describe them differently, which is the synonymy problem commonly found in tagging systems [2]. For this reason, semantically-related terms end up in different positions on the list, so it is difficult to recognize inter-related subjects and research areas.

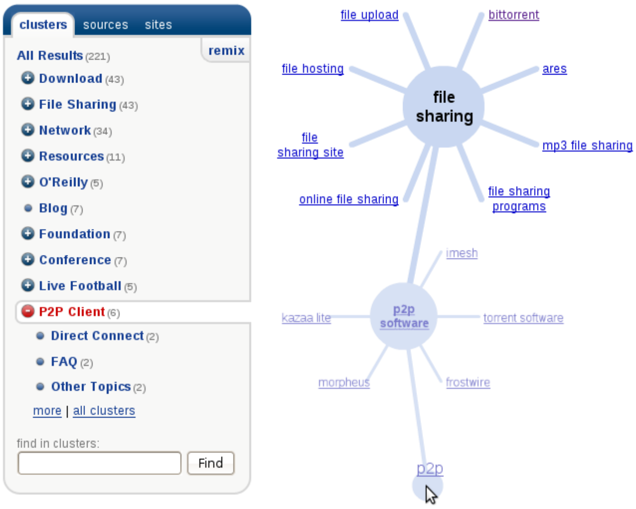
Although these cases are related to different communities, the practice of browsing and scanning many pieces of information to find relevant content is a very common issue. In both cases there are difficulties related to the query formulation, i.e. one only recognizes a relevant result when they go through it. In the case of the web forum, relevant results are posts, while in the case of the institute website, relevant results are professors or researchers with a shared goal or interest. We need a tool to visualize user-generated annotations which is able (a) to differentiate general and specific terms into different levels and (b) to provide a refinement mechanism which allows a user to browse horizontally, i.e. between different topics and, at the same time, vertically, i.e. doing an in-depth analysis and looking for specific terms.

## 3. BROWSING INTERFACES

Representing different levels of abstraction without polluting the interface is a challenging design task. A common way is to represent each level using indentation, e.g. the

[1] http://ubuntuforums.org

Clusty[2] search interface (Fig. 2a). The problem with this approach is that the user usually needs to scroll the page as he/she explores the structure, which requires an extra effort to keep the focus on a given abstraction level, i.e. a term and its proximate relationships. An efficient visualization technique which attempts to address this issue can be found in Google's Wonder Wheel[3] (Fig. 2b). It is a good example of a *focus & context* interface, which encompasses visualization techniques that allow a user to center his view on a part of the screen that is displayed in full detail (*focus*), while at the same time perceiving the wider screen surroundings in a less detailed manner (*context*). The major advantage of using these techniques is the improved space-time efficiency for the user, i.e. the information displayed per screen area unit is more useful and, consequently, the time required to find an item of interest is reduced as it is more likely to be already displayed [4].



(a) (b)

**Figure 2: (a) Clusty navigation menu and (b) Google Wonder Wheel.**

Fig. 2b shows an example of interaction. Let us suppose that the user is interested in downloading a peer-to-peer client, so he/she starts entering the query *"p2p"*. A new set of query suggestions appears and gains focus. Then, after selecting the *"p2p software"* query suggestion, the user is presented with new suggestions, among which is the query *"file sharing"*. Incidentally, the user may shift the search context and receive suggestions such as *"file hosting"* and *"file upload"*. If the purpose is to browse horizontally, the tool is very appropriate, leading the user to distinct domains with some level of specificity. However, the tool excludes the initial input term *"p2p"* from the query and eventually moves the users away from their search goal. Because the tool suggests related queries, it does not work as a query formulation tool. In other words, it does not necessarily keep all previously selected terms, suggesting queries that may not have a semantic intersection with the previous interactions.

## 3.1 Kolline's interface

Our solution consists of an interactive tool for visualizing hierarchies of user-generated annotations. Terms are related using an ontology which is, in turn, derived semi-automatically by applying a probabilistic model similar to the one presented in [6]. In the case of Ubuntu Forums, we extracted both text corpora and user-generated tags from threaded discussions. As for the institute website, we expect to gather patterns of term co-occurrence from researchers' papers. The resulting ontology is a hierarchy, in such a way that the closer a term is to the root, the more general it is. The purpose of our interface is to allow the user to browse this hierarchy, at first selecting general terms and then refining the search context progressively by adding more specific terms. Fig. 3 depicts Kolline's interface and highlights the functionality of our query formulation tool.

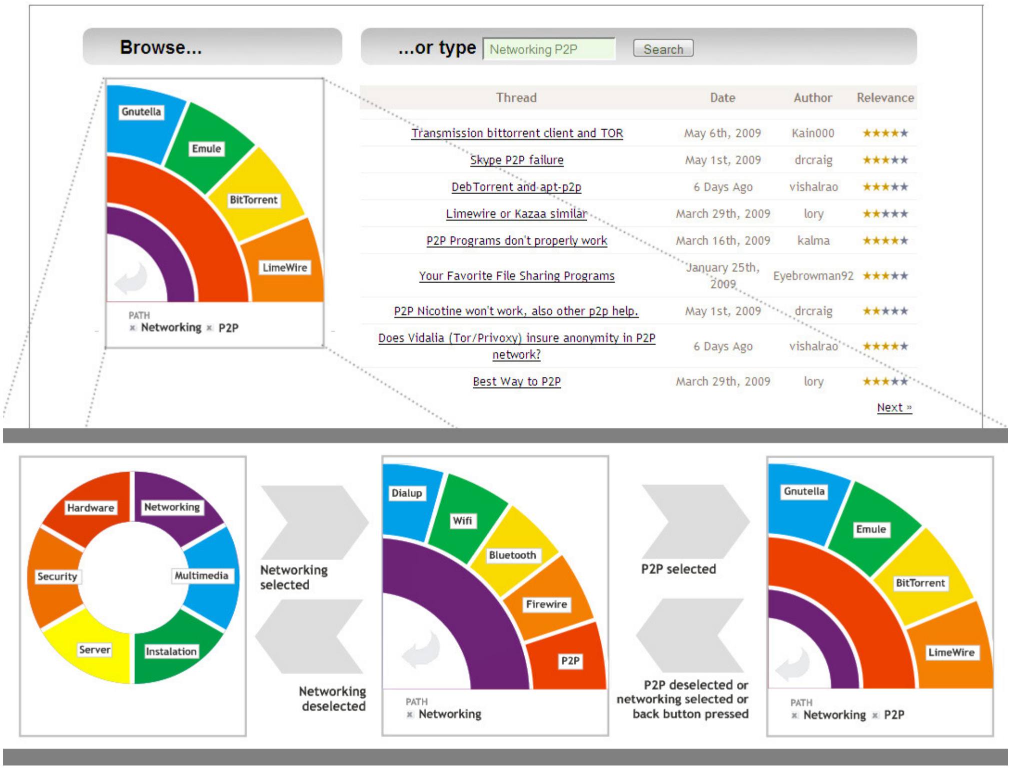
The design of the query formulation tool is based on a colored pie and each slice represents a term in the ontology. The scheme was inspired by an electronic memory game popularized in the eighties called Simon. The main goal of this game was memorizing the sequence of colors displayed by the interface, adding to the sequence one color at a time. In our design, the colors have the purpose of enhancing the user's working memory. [9] shows that recognition memory is 5%–10% better on colored images in comparison to black & white images. Thus, one important design decision is based on the idea that colors may have an important role on helping the user to memorize previous steps when interacting with the interface.

Another important design decision is to avoid scrolling. [7] points out that this approach provides a better experience, especially for novice users. In both cases shown in Fig. 2, the structure grows vertically as the user browses the interface. As a result, scrolling eventually becomes a required effort during the interaction. To address this issue, our query formulation tool stays static and within a single, limited area of the screen, showing just the two previously selected levels as inner circles, i.e. context, and new term recommendations in the outer circle, i.e. focus. The path below the quadrant shows all previously selected terms and allows the user to go directly to a certain level. This has an important role in keeping the user's attention on the focus, without loosing the visual contact of the context.

The tool works as follows (Fig. 3). On selection of one of the general terms displayed by the interface, a transition changes the tool's shape. It becomes a quadrant through a smooth transition to transmit the idea of changing the focus. Each previous level of the hierarchy, i.e. inner circle, keeps the color of the previously selected term. At each new selection, new semantically-related terms are recommended in the outer side of the quadrant. The user can move the mouse over the inner circles to view the context, which causes the previously selected terms to be highlighted. Each new interaction with the tool changes the remaining parts of the interface. The search box is automatically updated with the effective expression resulting from the user's selection. Newer selections refine the search results which in turn gives an instant feedback, so the user can make a decision to continue refining the search context or to go back and browse horizontally over the ontology. To go back, the user can click: (a) on the back arrow displayed near the center of the quadrant; (b) on an inner circle or (c) on a previously selected term in the path below the quadrant.

[2] http://clusty.com

[3] At the time of writing this paper, one could reach the tool by selecting "Show options" in the Google's main page.



| Thread | Date | Author | Relevance |
|---|---|---|---|
| Transmission bittorrent client and TOR | May 6th, 2009 | Kain000 | ★★★★☆ |
| Skype P2P failure | May 1st, 2009 | drcraig | ★★★☆☆ |
| DebTorrent and apt-p2p | 6 Days Ago | vishalrao | ★★★☆☆ |
| Limewire or Kazaa similar | March 29th, 2009 | lory | ★★☆☆☆ |
| P2P Programs don't properly work | March 16th, 2009 | kalma | ★★★★☆ |
| Your Favorite File Sharing Programs | January 25th, 2009 | Eyebrowman92 | ★★★☆☆ |
| P2P Nicotine won't work, also other p2p help. | May 1st, 2009 | drcraig | ★★☆☆☆ |
| Does Vidalia (Tor/Privoxy) insure anonymity in P2P network? | 6 Days Ago | vishalrao | ★★★★☆ |
| Best Way to P2P | March 29th, 2009 | lory | ★★★☆☆ |

Next »

**Figure 3: Kolline interface on top and a graphical representation of successive interactions on bottom.**

## 4. FUTURE WORK AND CONCLUSION

As for the evaluation, we will conduct a user study in two phases. First, we want to identify the strategy used by users when performing search tasks and observe their browsing practices. We are particularly interested in better understanding the users' main difficulties when formulating queries and identifying relevant results. Subjects will be recruited to participate in individual, moderated sessions. A screen capturing software will record user activity and moderator will take notes. The aim of the second phase is to assess Kolline's effectiveness in comparison with the tools regularly used by users for searching. For this purpose, a comparison test will be conducted and a group of participants will be asked to perform a set of predefined tasks, in a between-subjects design.

This paper presented a query formulation tool which employs visualization techniques for browsing. We analyzed two cases which involve user-generated annotations to classify content and described two examples of browsing interfaces that attempt to provide assistance to the user in information-seeking tasks. Our design decisions are aimed at addressing the problems found in the case studies and at dealing with the issues identified in usual web browsing interfaces. Therefore, our interface differentiates general and specific terms into different levels and provides a refinement mechanism which allows a user to browse horizontally and vertically over large ontologies.

## 5. REFERENCES

[1] G. Bowker and S. Star. *Sorting Things Out: Classification and Its Consequences.* MIT Press, Cambridge, MA, 1999.

[2] S. Golder and B. Huberman. Usage patterns of collaborative tagging systems. *Journal of Information Science*, 32(2):198–208, 2006.

[3] T. O'Reilly. What is web 2.0 | o'reilly media. Available online: `http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html`. Last access: 8/21/2009, September 2005.

[4] J. Porto de Albuquerque, H. Isenberg, H. Krumm, and P. L. de Geus. Improving the configuration management of large network security systems. *DSOM'05: Proc. of the 16th IFIP/IEEE International Workshop on Distributed Systems: Operations and Management*, pages 36–47, October 2005.

[5] A. H. Renear and C. L. Palmer. Strategic reading, ontologies, and the future of scientific publishing. *Science*, 325:828–832, August 2009.

[6] M. Sanderson and B. Croft. Deriving concept hierarchies from text. *SIGIR '99: Proc. of the 22nd international ACM SIGIR conf. on research and development in information retrieval*, pages 206–213, 1999.

[7] E. Schwarz, I. Beldie, and S. Pastoor. Comparison of paging and scrolling for changing screen contents by inexperienced users. *Human factors*, 25(3):279–282, 1983.

[8] J. Sinclair and M. Cardew-Hall. The folksonomy tag cloud: when is it useful? *Journal of Information Science*, 34(1):15, 2008.

[9] F. Wichmann, L. Sharpe, and K. Gegenfurtner. The contributions of color to recognition memory for natural scenes. *Learning, Memory*, 28(3):509–520, 2002.

# *Cognitive Dimensions* Analysis of Interfaces for Information Seeking

Gene Golovchinsky
FX Palo Alto Laboratory, Inc.
3400 Hillview Ave, Bldg. 4
Palo Alto, CA 94304 USA

gene@fxpal.com

## ABSTRACT

Cognitive Dimensions is a framework for analyzing human-computer interaction. It is used for meta-analysis, that is, for talking about characteristics of systems without getting bogged down in details of a particular implementation. In this paper, I discuss some of the dimensions of this theory and how they can be applied to analyze information seeking interfaces. The goal of this analysis is to introduce a useful vocabulary that practitioners and researchers can use to describe systems, and to guide interface design toward more usable and useful systems.

## Categories and Subject Descriptors

H.5.2 [**User Interfaces**]: Evaluation/methodology

## General Terms

Human Factors

## Keywords

Cognitive Dimensions, Information seeking, user interfaces, evaluation

## 1. INTRODUCTION

Cognitive Dimensions is a technique for analyzing complex information artifacts, including programming languages, device interfaces, and interactive software user interfaces.[3][5] It is designed to be a meta-analysis, a broad-brush approach that looks at structural aspects of the system and identifies characteristics that may impede or enable certain kinds of interactions with the system. It can be used in a summative or formative manner to evaluate existing systems and to drive design of new systems.

In this paper, I apply this tool to the domain of user interfaces for information seeking and exploration. This domain is characterized by complex, cognitively-rich activities. To be effective, information seeking tools need to be designed in a manner consistent with people's cognitive abilities: interfaces that work with people's strengths can be effective even when driven by relatively simple indexing and retrieval schemes; conversely, powerful retrieval engines can be made less usable by coupling them to awkward or ill-designed interfaces.

## 2. SEARCH INTERFACES

Information seeking is an inherently difficult activity due to a number of factors: peoples' information needs are often ill-defined, [1] they may lack the vocabulary required to express the information need, [10] and the need may evolve over time as new information is identified. [7]

These characteristics impose requirements on interfaces through which people look for information. To be useful, interfaces have to be simple to avoid burdening the searcher with distracting or unnecessary complexity, but not too simple to support the cognitive tasks characteristic of information seeking.

## 3. COGNITIVE DIMENSIONS

Cognitive Dimensions is an analytic tool that focuses on the process of interaction, rather than on static analysis of artifacts. [4] It differs from other analytic approaches such as GOMS/KLM [6] in that Cognitive Dimensions does not require a specialized, detailed, and time-consuming analysis that is predicated on very specific interface characteristics. Instead, it allows an interface to be discussed and compared with alternatives using broad terms, represented as different dimensions. The full theory identifies a large number of these dimensions, but only some of them are useful for most analyses of interactive information seeking systems. These dimensions will be discussed below; see [3] for a detailed discussion of all dimensions.

It is important to note that although the dimensions reflect different aspects of interaction, they are not completely orthogonal. In practice, this means that an interface flaw may be reflected simultaneously in more than one dimension.

### 3.1 Premature commitment

This dimension reflects the sequence of steps that a user must perform to achieve a specific outcome. If the user must make a decision early on in some interaction without necessarily having all information to understand the choice, we classify that as premature commitment. For example, being required to provide personally-identifying information prior to being able to interact with a system even in a light-weight manner is an example of premature commitment. So is forcing a user to click on a link in a search result to see some critical piece of information such as the price or an abstract.

Requiring people to ask the system for information that might have just as easily been shown right away was shown to reduce the use of that information. [11] Applied to information retrieval, this suggests that search results should include enough metadata that might help people assess the utility of the document, and accounts for the popularity of snippets as a way of explaining search results. There are limits, of course, to how much information can be presented for each result without making it

In Proceedings of the Third Workshop on Human-Computer Interaction and Information Retrieval (HCIR '09), October 23, 2009.

difficult for the user to understand the information, but designers should consider the tasks that cause people to search, and what information about specific results would make it easier to assess relevance or utility.

## 3.2 Viscosity

Viscosity assesses a design's resistance to change. If, for example, an interface requires the searcher to go through a series of menus or dialog boxes to switch between author search and content search, we say that the design has high viscosity. It is particularly unfortunate if high viscosity is coupled with premature commitment: the user is required to make choices without fully understanding the consequences and it is the difficult to undo these actions once additional information is learned.

Automatic query expansion based on recent browsing history (e.g., [1]) can generate viscosity as the system learns associations between terms and people that may outlast the utility of the association for a particular individual.

Yelp!'s faceted search interface (www.yelp.com) offers another example of unnecessary viscosity: A query can be formulated by selecting relevant facets, but once an item is selected, the facet information goes away, forcing the user to backtrack to revise the query. If the original design does not support good viscosity, it may be hard to introduce it later, although there may be measurable benefits to doing it. [8] Section 4 offers a more detailed analysis of the Yelp! interface.

## 3.3 Hidden dependencies

This dimension assesses the presence of hidden links among components of the system whose existence may be hard for users to learn. If these links impact people understanding of important system functions, the design should be rated unfavorably on this dimension. For example, "personalized search" learns from users' [9] or groups' [1] interactions with search results about their preferences, and uses that personalization information to affect search rankings. While this approach may improve precision, it may become progressively more difficult to understand why a particular document was or was not retrieved in response to a given query.

This dimension is also related to viscosity. In this case, however, the interface may not reflect to the user that a prior action is now affecting system responses, and it may not be obvious how (or even possible) to undo the effects of hidden dependencies.

## 3.4 Visibility

Visibility reflects how easy it is to view the various aspects of the system. It is related to the notion of affordance. A deep menu system may exhibit poor visibility. For example, Google's Trends search may generate output automatically in response to certain queries (cf. consistency, below) but if the searcher knows that they want to perform a search on structured data, they need to either have to remember the name and search for it, or they need to navigate a deep menu hierarchy (more/even more/labs/Google Trends) to discover the right place to search.

Poor visibility is particularly problematic with faceted search if the user cannot easily add, remove, or refine facet specifications.

## 3.5 Consistency

This is an obvious measure of the degree of similarity of means of accomplishing similar goals in different parts of the interface. It applies to layout (e.g., where people look for the search interface on a web site, where facets selection lists are located, etc.) and to the availability of features. The previous Yelp! example shows a degree of inconsistency because the query refinements are not available on a details page of a search result. Medynskiy et al [8] describe other challenges to making that interface more consistent.

## 3.6 Hard mental operations

Operations that rely on a user's concentrated attention may pose usability problems, particularly when a user may not have the right background knowledge to perform the operation, or may be operating with divided attention. Wolfram|Alpha's minimal interface that requires users to enter syntactically-complex queries is a good example of hard mental operations; Boolean query interfaces (notorious for being error prone for a variety of reasons) are a good example of hard mental operations.

Occurrences of hard mental operations may be exacerbated by high viscosity or premature commitment situations where the user may find it difficult to know what to do or what to undo when an error or unexpected result is observed.

## 3.7 Role-expressiveness

This dimension reflects how well the various visual components of an interface reflect their purpose and the operations available on them. Can the user find the search box? Is it obvious how to compare documents?

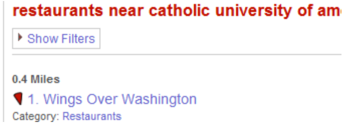
While common controls have become reasonably standardized, less common tools such as query expansion or certain kinds of faceted search may require more attention from the interface designer to make the purpose of the controls and the manner in which they should be used obvious, particularly when they are intended to support activity that may involve hard mental operations. The interface should strive for transparency rather than being cluttered with many controls that are used infrequently or whose purpose is not immediately clear.

## 3.8 Progressive evaluation

How easy is it for people to assess what they've discovered, how much progress they've made toward their goal? This dimension becomes particularly important for exploratory search. Interfaces that make it difficult to see which documents have been 'saved' or bookmarked, or ones that hide users' query history, requiring additional interaction to see what has been done may hamper people's exploratory search activities, because being able to get an overview of what has been found or query tactics that have been used may be a useful tool for assessing progress toward satisfying the information need that motivated the search in the first place.

# 4. AN EXAMPLE

You are planning to attend what promises to be an excellent workshop in the DC Area, and would like to plan ahead for some nice meals. The Yelp site offers a large listing of restaurants in the area, so it is an obvious choice to start looking. A search for "restaurants near catholic university of america" produces a map and a list of four restaurants: one has a review, two are fast food chains, and the fourth is called "Capital City Rehab Center." Clearly the query needs to be refined, for which purpose the site offers two possibilities: the ambiguously labeled "show filters" link, and a link labeled "Mo' Map" (Figure 1 and Figure 2).


restaurants near catholic university of am

▸ Show Filters

0.4 Miles

1. Wings Over Washington

Category: Restaurants

**Figure 1. "Show Filters" link**

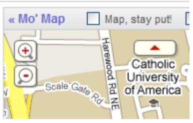
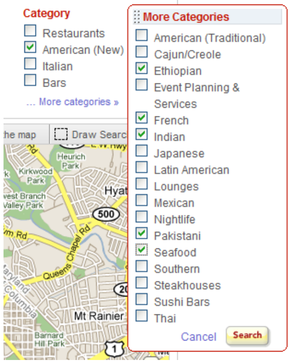


**Figure 2. "Mo' Map"**

Zooming out on the map increases the number of available restaurants to 121, a nice example of low *viscosity*. To refine the search, however, you need to know that "show filters" will allow you to select restaurants by cuisine, price, *etc*. "Filters' is an overly-technical term that has poor *role expressiveness* and *visibility*. Giving a few examples (*e.g.*, "by price", "by cuisine," etc.) would make it easier to transition to the next query refinement stage.

The filter interface lets you specify distance, features, price, and category, in addition to expanding to other cities and sorting the results. As you scroll down the list (which, with the distance expanded to two miles, now has 1695 entries), the map moves with you, again illustrating low *viscosity*. When price and features are specified, the category list is updated automatically, although only showing four items by default requires an extra step to see more categories (Figure 3).


Category

☐ Restaurants
☑ American (New)
☐ Italian
☐ Bars

... More categories »

More Categories

☐ American (Traditional)
☐ Cajun/Creole
☑ Ethiopian
☐ Event Planning & Services
☑ French
☑ Indian
☐ Japanese
☐ Latin American
☐ Lounges
☐ Mexican
☐ Nightlife
☑ Pakistani
☑ Seafood
☐ Southern
☐ Steakhouses
☐ Sushi Bars
☐ Thai

Cancel Search

**Figure 3. Selecting categories**

With the selected cuisines, the list is reduced to 75 restaurants. The sort order is not apparent: there is no indication in the interface, and casual inspection of the top five results rules out distance, number of reviews, and ratings. That leaves "Best Match" as the only seemingly possible order criterion, but it is not clear what that means. Thus we would classify this as a n example of a *hard mental operation* that affects the transparency of results.

The information displayed for each entry in the list also shows high *viscosity* and poor *visibility* because even though the price was restricted, the actual values for the retrieved restaurants were not shown. Instead, each link has to be interrogated individually to get that information. Nor does the system allow results to be grouped by price or by category, requiring the filter to be modified instead. This is another example of high *viscosity*.

It appears impossible to save promising restaurants in an *ad hoc* manner to generate a short list to pick from at the end of the search. This shows poor *progressive evaluation*, requiring some external record of promising locations.

Selecting a restaurant page creates a new set of usability challenges: although the map is still shown, it no longer displays other matching restaurants, although now it allows restaurants to be "bookmarked," showing high *viscosity* and poor *consistency*. It should be possible to bookmark a restaurant in any view. *Viscosity* is even worse when the back button is pressed, because the filter changes from a two-mile to a five-mile radius, increasing the number of matches and changing the size and scale of the map.

Yelp offers a link to browse nearby restaurants, but completely forgets the filters that had been set up just before. Instead, it shows a full list of available cuisines in the Zip code of the selected restaurant, but this view does not allow multiple categories to be selected, again demonstrating high *viscosity* and poor *consistency*. The "show filters" link is available below this list of categories, but it is not opened, requiring additional interaction from the user to refine the query. Furthermore, it no longer offers a category aspect for multiple selection, and offers "San Francisco" as a possible city to search. When the filter is engaged to re-create the original query, the list of cuisines persists, showing all cuisines available in the DC area, rather than the version available initially (Figure 3). The lack of *consistency* here is staggering.

This example illustrates several usability problems encountered during a short search session. It is by no means a definitive usability analysis of the Yelp! site, and is meant only to show the flavor of cognitive dimensions analysis. Although none of the problems identified above is critical, they do, in combination, affect the quality of the search interaction and may cause people to miss useful results or to repeat themselves. In more mission-critical or time-sensitive situations, these interface problems can contribute to more costly mistakes than not finding a great place to have dinner.

## 5. CONCLUSIONS

T.R.G. Green's Cognitive Dimensions Theory offers an interesting and powerful toolbox that can be used to characterize and reason about search interfaces without descending into the minutia of particular designs. The vocabulary of cognitive dimensions can form an effective shorthand for expressing complex characteristics of interfaces and systems, and therefore can improve communication between designers, system builders, and other stakeholders. While it was designed for broad applicability to information artifacts of all kinds, it is particularly useful for characterizing the kinds of complex systems that people are using to fulfill their information needs.

## 6. REFERENCES

[1] Balfe, E. and Smyth, B. 2004. Query Mining for Community Based Web Search. In *Proceedings of the 2004 IEEE/WIC/ACM international Conference on Web intelligence* (September 20 - 24, 2004). Web Intelligence. IEEE Computer Society, Washington, DC, 594-598. DOI= http://dx.doi.org/10.1109/WI.2004.120

[2] Belkin, N. J.; Oddy, R. & Brooks, H. 1982. ASK for Information Retrieval. Journal of Documentation, 38, 61-71 (part 1) & 145-164 (part 2)

[3] Green, T. R. G. 1989. Cognitive dimensions of notations. In A. Sutcliffe and L. Macaulay (Eds.) *People and Computers V*. Cambridge: Cambridge University Press, pp. 443-460

[4] Green, T.R.G. and Blackwell, A. 1998. Cognitive Dimensions of Information Artefacts: a tutorial. Available online at http://www.ndirect.co.uk/~thomas.green/workStuff/Papers/

[5] Green, T. R. G. 2000. Instructions and descriptions: some cognitive aspects of programming and similar activities. Invited paper, in Di Gesù, V., Levialdi, S. and Tarantino, L., (Eds.) Proceedings of Working Conference on Advanced Visual Interfaces (AVI 2000). New York: ACM Press, pp 21-28.

[6] John, B. and Kieras, D. E. 1996. The GOMS Family of User Interface Analysis Techniques: Comparison and Contrast, *ACM Transactions on Computer-Human Interaction 3,4*, 320-351.

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

[8] Medynskiy, Y., Dontcheva, M., and Drucker, S. M. 2009. Exploring websites through contextual facets. In *Proceedings of the 27th international Conference on Human Factors in Computing Systems* (Boston, MA, USA, April 04 - 09, 2009). CHI '09. ACM, New York, NY, 2013-2022. DOI= http://doi.acm.org/10.1145/1518701.1519007

[9] Teevan, J., Dumais, S. T., and Horvitz, E. 2005. Personalizing search via automated analysis of interests and activities. In *Proceedings of the 28th Annual international ACM SIGIR Conference on Research and Development in information Retrieval* (Salvador, Brazil, August 15 - 19, 2005). SIGIR '05. ACM, New York, NY, 449-456. DOI= http://doi.acm.org/10.1145/1076034.1076111

[10] Torrey, C., Churchill, E. F., and McDonald, D. W. 2009. Learning how: the search for craft knowledge on the internet. In *Proceedings of the 27th international Conference on Human Factors in Computing Systems* (Boston, MA, USA, April 04 - 09, 2009). CHI '09. ACM, New York, NY, 1371-1380. DOI= http://doi.acm.org/10.1145/1518701.1518908

[11] Wright, P. 1991. Cognitive overheads and prostheses: some issues in evaluating hypertexts. In *Proceedings of the Third Annual ACM Conference on Hypertext* (San Antonio, Texas, United States, December 15 - 18, 1991). HYPERTEXT '91. ACM, New York, NY, 1-12. DOI= http://doi.acm.org/10.1145/122974.122975

# Cognitive Load and Web Search Tasks

Jacek Gwizdka
Dept. of Library and Information Studies, School of Communication and Information, Rutgers University
New Brunswick, NJ, 08901 USA
HCIR2009@gwizdka.com

## ABSTRACT

Assessing cognitive load on web search is useful for characterizing search system features, search tasks and task stages with respect to their demands on the searcher's mental effort. It is also helpful in examining how individual differences among searchers (e.g. cognitive abilities) affect the search process and its outcomes. We discuss assessment of cognitive load from the perspective of primary and secondary task performance. Our discussion is illustrated by results from a controlled web search study (N=48). No relationship was found between objective task difficulty and performance on the secondary task. There was, however, a significant relationship between search task stages and performance on the secondary task.

## Categories and Subject Descriptors

H.1.2 [**Models and Principles**]: User/Machine Systems – *human information processing* H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *search process*

## General Terms

Measurement, Performance, Experimentation, Human Factors.

## Keywords

Cognitive load, search task, user behavior.

## 1. INTRODUCTION

Web search behavior is affected by the task, system, and individual searcher characteristics. These factors, either alone or in combination, influence the level of difficulty experienced by a searcher. One kind of difficulty is related to mental, or cognitive, requirements that are imposed by the search system or the task itself. Understanding what contributes to a user's cognitive load on search tasks is crucial to understanding the search process and to identifying search tasks types and search system features that impose increased levels of load on users. As new user interfaces and interactive features are introduced into the information search systems we need to understand how the new functionality affects user performance and the system usability, usefulness, and acceptance. For example, user relevance feedback is a feature that was reported to be avoided by users due to the heightened cognitive load [1].

In the next section we briefly discuss cognitive load and provide a short overview recent research that used cognitive load in the context of information search. We then highlight our results demonstrating that mental effort varies across search task stages.

## 2. BACKGROUND

The concept of cognitive load has been used in various fields that deal with the human mind interacting with external stimuli (e.g., ergonomics, psychology, learning). In this paper, we define cognitive load can as the mental effort required for a particular person to complete their task using a given system. Hence, at any point in task performance, cognitive load is relative to the user, the task being completed, and the system employed to accomplish the task.

It should be clear that cognitive load is of interest to interactive information retrieval researchers for two reasons. First, it can be used to characterize search interfaces with respect to cognitive cost. Second, it can be used to characterize user tasks and their elements with respect the required mental effort. Both perspectives have a long history in human factors and human-computer interaction literature. Most recently, the first approach was elaborated by Wilson and schraefel at the last year's HCIR workshop [16]. Wilson and schraefel proposed Cognitive Load Theory [4] as a tool useful in estimating cognitive costs of information search interfaces and proposed an inspection-based evaluation framework [18]. In other related recent work that exemplifies the first approach, Harper and colleagues established web page ranking according to their perceived visual complexity and linked it with cognitive load [9].

In CLT terminology, the first approach deals mainly with *extrinsic load*, that is with the complexity imposed by search interface and system. The second approach, deals mainly with *intrinsic load*, that is with search task demands on user's cognitive resources. The primary goal of the first approach is to lower the extrinsic load so that user can commit more cognitive resource to the good *germane load* that facilitates task performance. The primary goal of the second approach is to understand better mental requirements of search tasks. A factor that often mediates the effects the task and the system are the user's cognitive abilities (e.g., [6]).

This paper promotes the second approach, and also considers selected cognitive abilities in addition to task performance factors.

## 2.1 Measurement of Cognitive Load

Methods used to date to assess cognitive load included searcher observation, self-reports (e.g., using questionnaires, think-aloud protocols, and post-search interviews), dual-task techniques [5], [11], and various approaches that employ external devices to collect additional data on users (e.g., eye-tracking, pressure-sensitive mouse and other physiological sensors [10]). The two latter groups of techniques have the advantage of enabling real-time, on-task data collection. However, use of external devices can be expensive and impractical. Hence, the promise of dual-task (DT) method that allows for an indirect objective assessment of effort on the primary task. Only few studies employed this method to assess cognitive load in online search tasks [12][5].

The dual-task technique measures directly instantaneous cognitive load at discrete points in time. The discrete values are typically used to calculate averages over time intervals of interest (i.e., during performance of a task or a task stage). The average values reflect the intensity of the load [13], [19]. The intensity is related to the overall load perceived by a person, but is not necessarily the same, as it is often assumed.

HCIR Workshop 2009, Washington, D.C., October 23 2009.

We present a study that employed the dual-task method as the technique for assessing cognitive load on web search tasks.

# 3. METHODOLOGY

The details of the experimental methodology were reported in [7], [8]. However, the results presented in this paper have not been reported earlier. This section provides only this information that is needed for understanding the main points.

Forty-eight subjects participated in a controlled web-based information search. Two cognitive abilities were assessed, working memory and spatial ability. The study search tasks were designed to differ in terms of their difficulty and structure. Two types of search tasks were used: Fact Finding (FF) - find one or more specific pieces of information, and Information Gathering (IG) - collect several pieces of information about a given topic. The tasks were also divided into three categories based on the structure of the underlying information need, 1) Simple (S), satisfied by a single piece of information; 2) Hierarchical (H), satisfied by finding multiple characteristics of a single concept (a depth search); 3) Parallel (P), satisfied by finding multiple concepts that exist at the same level in a conceptual hierarchy (a breadth search) [15]. Based on these characteristics, the tasks were categorized into three levels of "objective" difficulty. FF-S was assigned low difficulty level, FF-P and FF-H middle-difficulty level, and IG-H and IG-P high difficulty level. During the course of each study session, participant performed a set of six tasks of differing type and structure. The search tasks were performed on the English Wikipedia by using two search engines with the associated search interfaces: U1 Google, and U2 ALVIS [3]. The order of tasks was partially balanced with respect to the objective task difficulty to obtain all possible combinations of low-medium-high and high-medium-low difficulty within the groups of three tasks. This yielded four task rotations that were repeated for two orders of user interfaces. Thus there were eight task/UI rotations.

A secondary task (DT) was introduced to obtain indirect objective measures of user's cognitive load on the primary search task. A small pop-up window was displayed at a fixed location on a computer screen at random time. The pop-up contained a word with a name of a color. The color of the word's font either matched or did not match the name of the color. Participants' were asked to click on the pop-up as soon as they noticed it. The secondary task involved motor action, as well as visuo-spatial and verbal/semantic processing. The modalities of the primary task and the secondary task overlapped. One could have reasonably assumed that higher demands on cognitive resources by the primary search task would be reflected in lower performance on the secondary task.

## 3.1 Data Collection and the Measures

User interaction was recorder by Morae screen cam software from TechSmith and by the secondary task software. The interaction logs that were used in the analysis presented in this paper included time-stamped sequences of visited web pages, keyboard clicks, and mouse clicks. The latter were recorded for the primary and the secondary task.

User search process was divided into four main task stages (Figure 1). We used a semi-automatic process to segment user interaction data into task stages. The process involved classifying URLs, and detecting patterns in the keyboard and mouse data. The data collected for 48 users contained 288 tasks and 1447 task stages.

The two main controlled factors were the objective task difficulty (OBJ_DIFF) and the search system (UI). The additional two independent factors were the levels of working memory (WM) and spatial ability (SA). We assessed intensity of cognitive load within each task stage by calculating the average reaction time (RT) to the secondary task events.

# 4. RESULTS

The analysis presented in this paper focuses on the relationship between the independent variables and the performance on the secondary task (RT). The analysis was performed at the task and the task-stage levels.

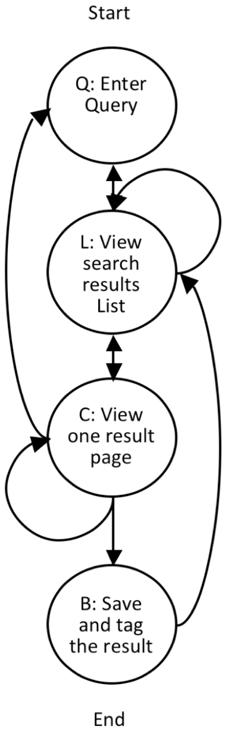


**Figure 1. State diagram of task stages.**

An analysis of covariance (ANCOVA) performed with the objective task difficulty, task stage and user interface as fixed factors and with the cognitive abilities as covariates revealed that mean reaction time differed significantly between task stages (F(3,862)=6.2, p<.001) and was significantly related to both cognitive abilities (F(1,862)=5.5, p<.05 for WM and F(1,862)=24.7, p<.001 for SA). There was no significant effect of the objective task difficulty or of the user interface. Post-hoc analysis (Bonferroni test) showed that the mean reaction times during the query and the bookmarking stage were significantly longer than during the search results list and the content stage (p<.05). Other differences were not significant. The differences in reaction time between the task stages are shown in Figure 2.

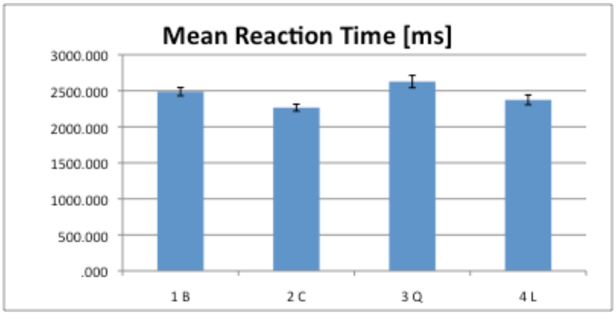


**Figure 2. Mean reaction time to the secondary task events.**

Reaction time could be considered as containing a component related to person (dependent on motor, perceptual, and cognitive ability of the person), a component related to task stage (dependent on the cognitive demands of the stage), and some other components (Equation 1).

$$RT_{total} = C + RT_{person} + RT_{task\_stage} \quad (1)$$

To further examine the two significant sources of variability in reaction time, individual user variability and task stage, we considered them separately. An analysis of variance was performed with participant identifiers as the main factor and reaction time as the dependent. The predicted mean values of reaction time and the residuals were then analyzed separately as dependents with other previously described independent factors. This statistical procedure is essentially equivalent to subtracting mean reaction time that is typical for a user in the given circumstances (the predicted value that represents $RT_{person}$) from the overall reaction time [13]. The resulting reaction time (the residual that represents $C + RT_{task\_stage}$) does not contain variability that could be ascribed to individual participants. This procedure removes the differences in the users' motor and cognitive skills. The expectation was confirmed by finding that the "typical" users' reaction time was significantly related to the users' cognitive abilities, while the residual reaction time was significantly related to the task stage. Clearly, the differences between task stages were reflected in different reaction times to the secondary task events.

Before we could draw final conclusions one more check needed to be performed. The secondary task involved using a mouse. The two task stages, during which the average reaction time was found to be the slowest (query entry and bookmarking/saving a relevant document), involved typically a fair amount of keyboard activity (query entry or tag entry). An additional check was thus performed to ensure that the longer reaction times were not related to the increased keyboard activity. Indeed, we found that the number of keystrokes and the time on keyboard were not related to the reaction time. We could conclude that the differences in reaction time among task stages were likely not due to motor activities.

## 5. SUMMARY & CONCLUSIONS

The aim of this paper was to present assessment of cognitive load on search tasks as a way of gaining better understanding of the web search process by characterizing it with the levels of cognitive load. We described user study that employed dual-task method as a technique for assessing cognitive load on web search tasks. The results showed that mental effort varied across search task stages. To our knowledge, this is the first study that demonstrated such an effect.

Our study also makes a methodological contribution. The results indicate that measures of cognitive load intensity may be sensitive to dynamic changes in task demands (such as the changes between task stages) and not sensitive to the differences between tasks. This finding explains why Schmutz and colleagues [14] and why our earlier analysis [7] of dual-task performance did not find significant relationships between reaction times and tasks.

Understanding mental effort imposed by task stages informs the design of search systems. It indicates, indirectly, during which stages the searchers may be more likely to have "spare" mental capacity and be willing to provide additional information to the system (e.g., relevance feedback). It could also be used in the design of notification delivery from other computing tasks [1].

Beyond the implications of specific results, the described method of assessing cognitive load can be applied in other web search contexts. It could be used in experiments designed to measure extrinsic load (related to the specific user interface), and, possibly, to corroborate results of evaluation frameworks such as the recently proposed approach [18].

## 6. ACKNOWLEDGMENTS

This research was sponsored, in part, by a grant from Rutgers University Research Council #RCG202130.

## 7. REFERENCES

[1] Adamczyk PD, Bailey BP. If not now, when?: The effects of interruption at different moments within task execution. Proceedings of CHI'2004.

[2] Back J, Oppenheim C. A model of cognitive load for IR: Implications for user relevance feedback interaction. Information Research 2001; 6(2). Reference available from: http://informationr.net/ir/6-2/ws2.html

[3] Buntine W, Valtonen K, Taylor M. The ALVIS Document Model for a Semantic Search Engine. Proceedings of the 2nd Annual European Semantic Web Conference; May 29, 2005. Heraklion, Crete.

[4] Chandler P, and Sweller J. Cognitive Load Theory and the Format of Instruction. Cognition and Instruction 1991; 8(4): 293-332.

[5] Dennis S, Bruza P, McArthur R. Web searching: A process-oriented experimental study of three interactive search paradigms. J Am Soc Inf Sci Tech 2002; 53(2): 120-133.

[6] Gwizdka J, Chignell M. Individual differences and task-based user interface evaluation: A case study of pending tasks in email. Interact Comput 2004; 16(4): 769-797.

[7] Gwizdka J. Assessing Cognitive Load on Web Search Tasks. (in press). To appear in The Ergonomics Open Journal. Bentham Open Access.

[8] Gwizdka J, Lopatovska I. The Role of Subjective Factors in the Information Search Process. To appear in Journal of American Society for Information Science and Technology (JASIST). Early access online 2009. DOI: 10.1002/asi.21183

[9] Harper S, Michailidou E, Stevens R. Toward a definition of visual complexity as an implicit measure of cognitive load. ACM Transactions on Applied Perception (TAP) 2009; 6(2): 1-18.

[10] Ikehara CS, Crosby ME. Assessing cognitive load with physiological sensors. Proceedings of the 38th Hawaii International Conference on System Sciences (HICSS); 3-6 January 2005. Big Island, HI, USA. IEEE Computer Society 2005. pp. 295a

[11] Kaki M. Findex: search result categories help users when document ranking fails. Proceedings of the ACM Conference on Human Factors in Computing Systems CHI; April 2-7, 2005. Portland, Oregon, USA. ACM Press; pp. 131–140.

[12] Kim YM, Rieh SY. Dual-Task Performance as a Measure for Mental Effort in Library Searching and Web Searching. Proceedings of the 68th Annual Meeting of the American Society for Information Science & Technology. Oct. 28 – Nov. 2, 2005. Charlotte, NC.

[13] Madrid IR, Van Oostendorp H, Puerta Melguizo MC. The effects of the number of links and navigation support on cognitive load and learning with hypertext: The mediating role of reading order. Comput Human Behav 2009; 25(1): 66-75.

[14] Schmutz, P., Heinz, S., Métrailler, Y. & Opwis, K. (2009). Cognitive Load in eCommerce Applications - Measurement and Effects on User Satisfaction. Adv in Human Comp Interact.

[15] Toms E, O'Brien H, Mackenzie T, et al. Task Effects on Interactive Search: The Query Factor. In: Fuhr N, Kamps J, Lalmas M, Trotman A. Eds. Focused Access to XML Documents. Lect Notes Comput Sci 4862. Springer Verlag 2008; pp. 359-372.

[16] Wilson ML, schraefel mc. Improving Exploratory Search Interfaces: Adding Value or Information Overload? In: Second Workshop on Human-Computer Interaction and Information Retrieval, 23rd October 2008, Redmond, WA, USA.

[17] Wilson ML, schraefel mc. Reading between the lines: identifying user behaviour between logged interactions. In: SIGIR09 Workshop: Understanding the User - Logging and interpreting user interactions in information search and retrieval, 23rd July 2009, Boston, MA, USA.

[18] Wilson ML, schraefel mc, White RW. Evaluating Advanced Search Interfaces using Established Information-Seeking Models. J Am Soc Inf Sci Tech 2009; 60 (7): 1407-1422.

[19] Xie, B., & Salvendy, G. (2000). Prediction of Metal Workload in Single and Multiple Task Environments. Int J Cog Erg, 4(3), 213-242.

# Visualising Digital Video Libraries for TV Broadcasting Industry: A User-Centred Approach

Mieke Haesen  Jan Meskens  Karin Coninx
Hasselt University - tUL - IBBT,
Expertise Centre for Digital Media,
Wetenschapspark 2, B-3590 Diepenbeek, Belgium
firstname.lastname@uhasselt.be

## ABSTRACT

Finding a suitable video fragment in a vast video archive is mostly a complex task. Even professional users have to skim many hours of stored video data before they find the desired content. In this paper, we present a user-centred software engineering approach that is employed to create a novel news video explorer for TV broadcasting industry. This approach helps to ensure the balance between the technological progress in the field of information retrieval on the one hand and the needs and goals of the end users on the other hand.

## Categories and Subject Descriptors

H.1.2 [**Models and Principles**]: User/Machine Systems—*Human Factors*; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*User-Centered Design*; H.3.3 [**Information Storage and Retrival**]: Information Search and Retrieval—*Information filtering, Selection Process*

## General Terms

Design, Human Factors

## Keywords

User-Centred Software Engineering, Searching and Browsing Video Archives

## 1. INTRODUCTION

In TV Broadcasting industry, professionals frequently have to search a vast video archive. Finding the desired video is often a difficult task using search engines for commercial or professional use. Moreover, retrieving a suitable fragment of a few seconds mostly requires users to skim many hours of stored video data using traditional video player software. In order to optimise this task, more advanced video browsers and visualisations are currently being employed in several research projects [2, 3, 6, 12, 14].

In this paper, a User-Centred Software Engineering (UCSE) approach is being employed to construct novel video information retrieval visualisations for the TV broadcasting domain. As a first step in this approach, TV researchers are interviewed and observed while they are working in their natural environment. This results in a better understanding of their practices and problems and helps taking the needs and goals of end users into account from the beginning of the process. By observing the end-users before any design takes place, the resulting visualisations can offer increased ease of use, efficiency and satisfaction [13].

The work in this paper is carried out within the context of the AMASS++ project [7]. This project aims to investigate the alignment and summarization of multimedia archives. The visualisations presented throughout this paper are built on top of the AMASS++ annotated news video corpus.

In summary, the major contributions in this paper are:

- a *UCSE process* employed in cooperation with a TV broadcasting company;
- an *interactive prototype for exploring news videos*, resulting from the aforementioned UCSE approach. The visualisations employed in this prototype are suited for TV researchers.

## 2. USER-CENTRED PROCESS

To provide suitable visualisations for the target group, we followed a user-centred software engineering approach. By involving end users from the beginning of the development process, it is more likely that the visualisation of the final user interface corresponds to their needs and goals [13]. The development process that is applied, is based on a framework for user-centred software engineering [4]. Figure 1 shows all stages of the process, including extracts of the artefacts that were used during each stage.

The end users involved in this development process are TV researchers. In the first stage of the process, where the *new system* is examined, a Contextual Inquiry (CI) is conducted in cooperation with the TV broadcasting company. A CI involves observing and interviewing end-users while they are performing their daily activities. This user study learned us that TV researchers need to browse large amounts of data. Their main job is to search video fragments of news broadcasts presenting particular people or situations that will be used in a TV programme or news broadcast. Based on keywords and other search criteria (e.g. date, programme title) the archive is searched to find suitable video fragments. Currently the videos in the archive are annotated

*HCIR* '09 Washington DC, USA

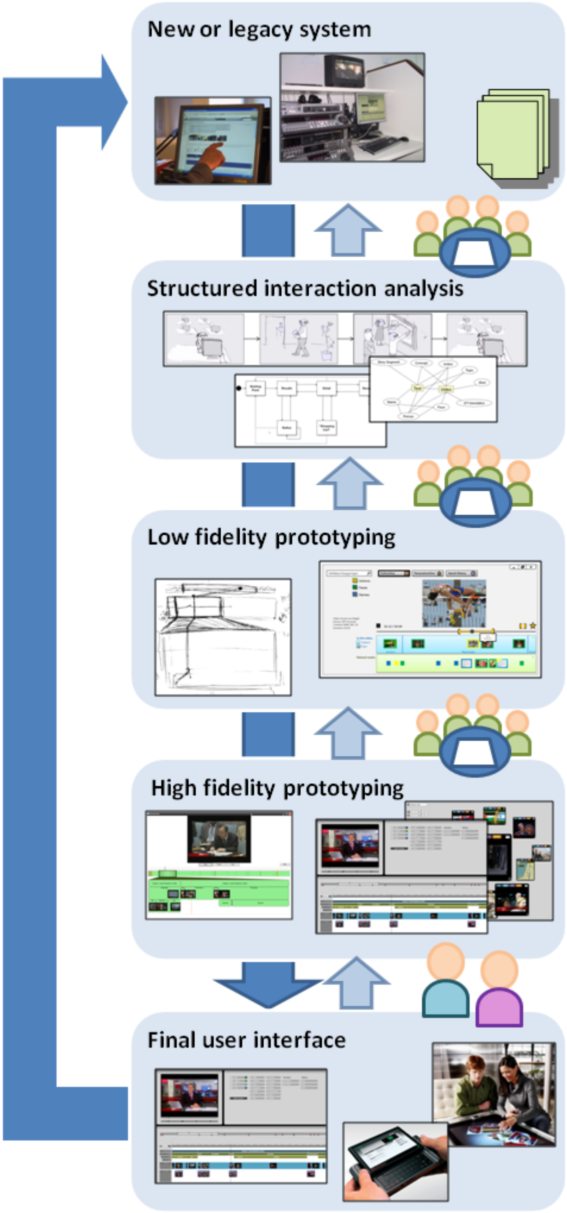
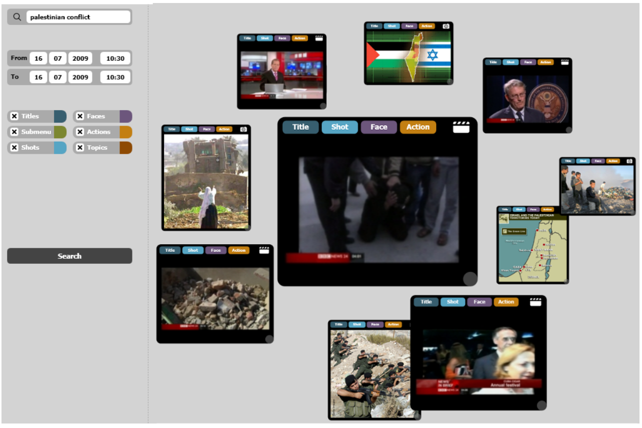


**Figure 1: The user-centred process that was adopted for the development of the user interface.**

manually, which allows the selection of appropriate archive videos. However, once a video is selected from the archive, the TV researcher has to browse the entire video manually in order to find and select a suitable fragment. Moreover, to carry out the different tasks, the user needs to combine several separate applications, which decreases efficiency.

The CI resulted into a scenario of use and an accompanying storyboard, that exemplify how one integrated future application can be used for searching archives, browsing an archive video and adding video fragments to a favorites folder. The storyboard was used to discuss the application with the stakeholders and provided the first data for the *structured interaction analysis*, in which a dialogue model and a conceptual model were created. Each artefact created in these early stages, was used for prototyping the UI.



palestinian conflict

From 16 07 2009 10:30

To 16 07 2009 10:30

**Figure 2: The search user interface.**

The first *low-fidelity prototypes* of the user interface were created using pencil and paper and Powerpoint. Following, the *high-fidelity prototypes* were created in .NET. During several iterations, the low- and high-fidelity prototypes were verified in stakeholder meetings, features were added gradually and a graphic designer was involved for the detailed UI design. The UI designs and visualisations, included in the high-fidelity prototype, are discussed in the following section.

## 3. NEWS VIDEO EXPLORER PROTOTYPE

During the aforementioned UCSE process, a news video explorer prototype was iteratively constructed. The user interface of this system contains two major parts: a *search user interface* and a *video browser*. While the former helps with finding the suitable video, the latter supports end users in skimming this video to locate and save content of interest.

### 3.1 Search User Interface

In order to find the right video, users start with entering a search query containing a keyword and/or date range in the search user interface (see Figure 2, left). Based on this information, the system retrieves a set of relevant news videos (see Figure 2, right). The location and size of each video thumbnail indicate the relevance with respect to the search query: the most relevant videos are bigger and located in the center of the screen. By replacing and resizing the video thumbnails, users can also sort the search results themselves.

Each video is represented by an animated slideshow of key-frames, which are computed by the shotcut detection algorithm described by Osian et al. [9]. When a video seems to be interesting, users can double click on it to open the video browser. We employed a keyframe-based abstraction technique since these have been shown to be effective in helping people quickly obtain a general understanding of what is contained in a video [2].

### 3.2 Video Browser

The video browser combines an advanced time slider, based on the time sliders in commercial video players such as Apple Quicktime Player and Microsoft Windows Media Player, with a timeline video visualisation [6]. A time slider is employed to manipulate the current time of the played video

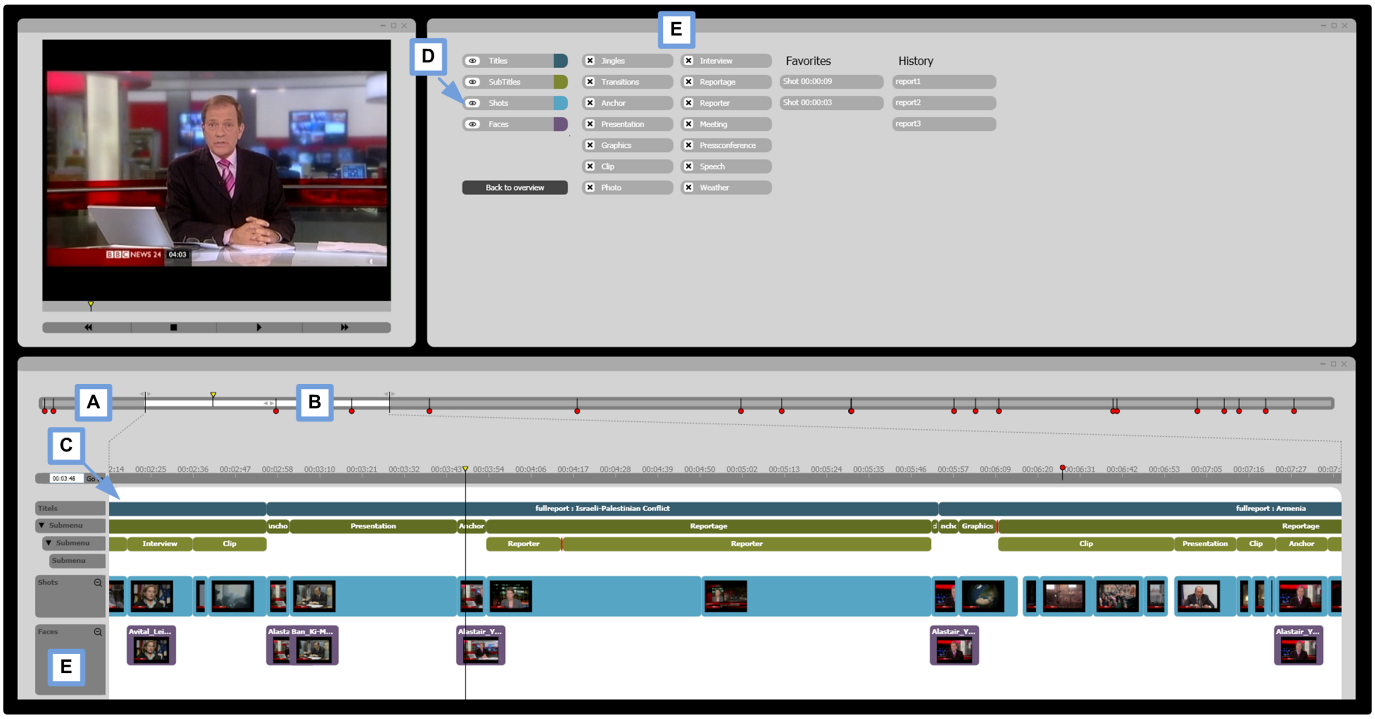


**Figure 3: The video browser.**

fragment (Figure 3, part A) and to specify an area of interest around this time (Figure 3, part B). The timeline (Figure 3, part C) gives a detailed view on the content in this area of interest.

The combination of the advanced video time slider with the timeline is in line with the basic idea of *focus+context* in information visualisation [1]. *Context*, on the one hand, is visualised using the video time slider, where red dots indicate the parts of the video relevant to the search query. *Focus*, on the other hand, is visualised in the timeline. Similar to other timeline based approaches [11], we use semantic zooming to specify the level of detail in the timeline. By resizing the focus area in the time slider (Figure 3, part C), users can zoom in or out on the video timeline. As shown schematically in Figure 4: a small focus area increases the level of detail in the timeline, a wider focus area decreases this level of detail.

The timeline shows a layered view on the video as computed by information retrieval algorithms for video and manual content annotations [7, 9, 10]. At the first layer, the title of every news item in this video is shown. This layer is further subdivided in several sublayers each containing story and scene information. The two remaining layers of the third timeline contain thumbnails of each shot in the movie and the names of the persons that appear in these shots. For example, Figure 3 shows that the topic about the Israeli-Palestinian conflict starts with the news anchor, followed by a presentation, the anchor again and a reportage. The faces layer reveals the names of the anchor (Alistair Yates) and presenter (Ban Ki-moon) together with a thumbnail.

The video browser contains several mechanisms to keep an overview on the large amount of information that is visualised in the timeline. Each layer can be maximised/minimised by clicking on its *toolglass icon*. In order to hide a layer, users can gray out the *eye buttons* (see Figure 3, part D), comparable to layers in traditional graphics software packages. Content filters (see Figure 3, part E) are provided to filter content from the timeline. For example, a user can check the *anchor filter* to remove the anchor blocks from the timeline. Users can also bookmark interesting blocks of content for later use.

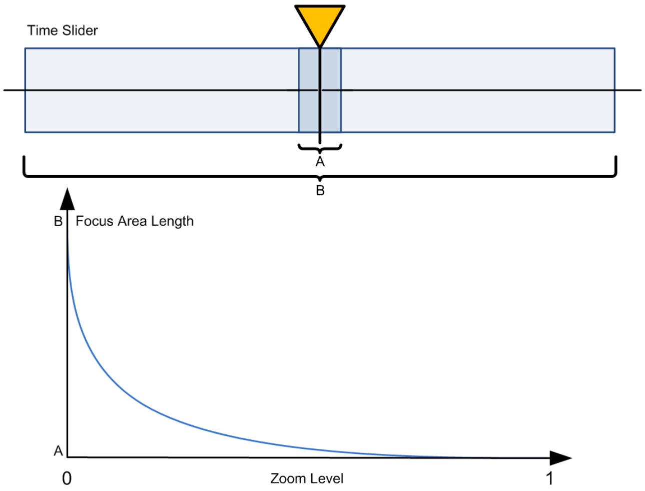


**Figure 4: The semantic zoom function.**

## 4. DISCUSSION AND FUTURE WORK

This paper presented a news video explorer for the TV broadcasting industry, realised by means of a UCSE process. First, the user tasks were observed and analysed, followed by the creation of structured interaction diagrams. Throughout the low- and high-fidelity prototyping stages, the structured interaction diagrams were used for verification of the prototypes. At several stages in the process, artefacts were discussed and verified in stakeholder meetings.

During the user-centred process, intermediate prototypes were frequently verified in stakeholder meetings. This resulted in interesting recommendations for the news video explorer presented in this paper. Although experts of the domain were involved in these meetings, thorough validation is needed to estimate the value of our news video explorer for TV researchers. Comparative and repeated user experiments could be helpful to improve our prototype and to discover the way in which the news video explorer can change daily practice for the TV researchers over time.

Our current prototype allows TV researchers to search and browse the video archive on their desktop pc. However, for meetings and assembling videos, they often have to move to other locations where it is not possible to consult the videos or different applications need to be used. In their current system, video files have to be saved on a central server in order to make them accessible on multiple PCs and locations. The use of modern devices such as multitouch tables or mobile devices might improve this approach. Therefore, we are currently investigating how the UI designs presented in this paper can be extended to other platforms such as a multitouch table and a ultra mobile pc. While a multitouch application is helpful for presenting and discussing archive videos during editorial meetings, a mobile application can assist journalists for carrying on particular videos or quick searches on location.

As indicated by the arrow on the left of Figure 1, targeting novel computing platforms is done by starting a new iteration of the user-centred process. Tool support [8, 5] for storyboarding and multi-device UI development will be investigated and deployed to provide smooth transitions between several stages in the user-centred process.

Besides video libraries, the AMASS++ project aims to provide technologies for cross-media and cross-language search and summarization in several application domains. Therefore, we will explore text based visualisations that allow users to quickly browse a text and its related multimedia. Additional visualisations of the search results, including map and timeline views, will also be considered here.

## 5. ACKNOWLEDGMENTS

The research described in this paper is based on the results of IWT project AMASS++ (SBO-060051). The authors would like to thank the consortium for their contributions made to this work and the companies of the user committee for their involvement in the user studies and their valuable feedback. Furthermore, we would like to thank Jimmy Cleuren for his contribution to the high-fidelity prototypes and Karel Robert for the graphic design of the user interface.

## 6. REFERENCES

[1] S. K. Card, J. D. Mackinlay, and B. Shneiderman. *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999.

[2] K.-Y. Cheng, S.-J. Luo, B.-Y. Chen, and H.-H. Chu. Smartplayer: user-centric video fast-forwarding. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 789–798, New York, NY, USA, 2009. ACM.

[3] M. G. Christel. Supporting video library exploratory search: when storyboards are not enough. In *CIVR '08: Proceedings of the 2008 international conference on Content-based image and video retrieval*, pages 447–456, New York, NY, USA, 2008. ACM.

[4] M. Haesen, K. Coninx, J. V. den Bergh, and K. Luyten. Muicser: A process framework for multi-disciplinary user-centred software engineering processes. In *HCSE '08: Second Conference on Human-Centered Software Engineering,*, pages 150–165, 2008.

[5] M. Haesen, K. Luyten, and K. Coninx. Get your requirements straight: Storyboarding revisited. In *Interact 2009 (to appear)*.

[6] A. Haubold and J. R. Kender. Vast mm: multimedia browser for presentation video. In *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 41–48, New York, NY, USA, 2007. ACM.

[7] S. Martens, J. H. W. Becker, T. Tuytelaars, and M.-F. Moens. Multimodal data collection in the AMASS++project. In *Multimodal Corpora: from Models of Natural Interaction to Systems and Applications*, pages 60–63. European language resources association, 2008.

[8] J. Meskens, J. Vermeulen, K. Luyten, and K. Coninx. Gummy for multi-platform user interface designs: shape me, multiply me, fix me, use me. In *AVI '08: Proceedings of the working conference on Advanced visual interfaces*, pages 233–240, New York, NY, USA, 2008. ACM.

[9] M. Osian and L. Van Gool. Video shot characterization. *Mach. Vision Appl.*, 15(3):172–177, 2004.

[10] P. T. Pham, M.-F. Moens, and T. Tuytelaars. Linking Names and Faces: Seeing the Problem in Different Ways. In *Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition*, Marseille France, 2008. Erik Learned-Miller and Andras Ferencz and Frédéric Jurie.

[11] C. Plaisant, B. Milash, A. Rose, S. Widoff, and B. Shneiderman. Lifelines: visualizing personal histories. In *CHI '96: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 221–ff., New York, NY, USA, 1996. ACM.

[12] B. T. Truong and S. Venkatesh. Video abstraction: A systematic review and classification. *ACM Trans. Multimedia Comput. Commun. Appl.*, 3(1):3, 2007.

[13] I. Wassink, O. Kulyk, E. van Dijk, G. van der Veer, and P. van der Vet. Applying a user-centred approach to interactive visualization design. In *Trends in Interactive Visualization.* Springer Verlag, London, 2008. ISBN=978-1-84800-268-5.

[14] M. Worring, C. Snoek, O. de Rooij, G. P. Nguyen, R. van Balen, and D. Koelma. Mediamill: Advanced browsing in news video archives. In *CIVR*, pages 533–536, 2006.

# Log Based Analysis of How Faceted and Text Based Search Interact in a Library Catalog Interface

Xi Niu
University of North Carolina at Chapel Hill
CB#3360, 100 Manning Hall
Chapel Hill, NC 27599-3360
xiniu@email.unc.edu

Cory Lown
North Carolina State University
NC State Library
Raleigh, NC 27695-7111
corylown@gmail.com

Bradley M Hemminger
University of North Carolina at Chapel Hill
CB#3360, 100 Manning Hall
Chapel Hill, NC 27599-3360
bmh@ils.unc.edu

## ABSTRACT

Faceted based search is an increasingly common part of search interfaces. This study examines the use of a library catalog search interface which supports but text searching and faceted based searching. Log analysis is performed of library catalog search records to analyze how and when faceted based searching is used in conjunction with text based searching. The logs are from the Triangle Research Libraries Network, which all use an Endeca based catalog search system. Results show that faceted based search is used much less frequently than text searching, and the usage clusters into certain categories of search behaviors.

## Categories and Subject Descriptors

H.1.2 [Information Systems]: USER/ MACHINE SYSTEMS—*Human information processing*

## General Terms

Human Factors, Measurement

## Keywords

Faceted based searching, text searching, library catalog, searching behavior, transaction log analysis, cluster analysis, Markov model, action pattern

## 1. INTRODUCTION

The purpose of this paper is to demonstrate a method for harvesting, storing, and analyzing data from the transaction logs of modern, faceted, search and browse Online Public Access Catalogs (OPACs). Faceted navigation OPACs, such as the ones in current use at the Triangle Research Libraries network (TRLN, comprised of the University of North Carolina, North Carolina State University, Duke, and North Carolina Central University), provide users with the ability to explore library collections by text searching and facet selection. Faceted search engines are emerging as the latest trend for search and navigation on library Online Public Access Catalogs (OPACs), as well as public libraries, WorldCat and general purpose search engines. Different from traditional OPACs, faceted search exposes the metadata (facets) that summarize records generated by the initial query as part of an interactive interface, allowing users to drill down along a particular dimension to desired results.

Since the faceted navigation catalogs are new, there is little data or literature available to suggest that these catalogs actually improve the user experience over traditional OPACs that offer only text searching. Through transaction log data analysis, this research aims at revealing the ways users interact with Endeca systems in UNC Library and find how people incorporate facets into their search process. In addition, it makes a methodological contribution of how we extract and process the transaction log data and how we apply some analytical methods (Markov stochastic modeling, cluster analysis) to the data to find search patterns for library catalog patrons.

## 2. RESEARCH QUESTIONS

The broad research question for this research is to investigate how people use facets in combination with text search in faceted library catalog. Under this broad question, we have three sub-questions: 1) How do people use facets to help them in their search process? Is it only as a refinement? Is text searching the assumed starting point for most faceted searches? How frequently do they use faceted search? 2) Do search sessions naturally segregate into certain types of search patterns that are discernable by log analysis? 3) When facets are utilized during the search process, are there any typical action sequences (patterns) commonly seen?

The first sub-question can mostly be answered by the descriptive statistics from the log data. The second and the third sub-questions require additional analytical methods to resolve.

## 3. RESEARCH METHODS

### 3.1 Data Extracting and Processing

While records from all the TRLN libraries are being analyzed as part of this project, the data reported in this paper are from the UNC library catalog over the period of Dec 16, 2008 to April 2, 2009. Over 1,200,000 query records logged by UNC Library Apache Server were cleaned and processed using Perl scripts and a MySQL database. Perl scripts were used to filter, parse, group and code the raw log. The MySQL database was used primarily as a way of sorting the data according to a particular variable and also joining two datasets based on a particular field.

A typical transaction record looks like this (on a single line):

```
71.70.185.34 - - [09/Mar/2009:00:13:53 -0400] "GET
/search?Ntk=Keyword&Ne=2+200043+206475+206590+11&N=206432&Ntt
=boston+globe HTTP/1.1" 200 40035
"http://search.lib.unc.edu/search?Nty=1&Ntk=Keyword&Ntt=boston+globe"
"Mozilla/5.0 (Macintosh; U; PPC Mac OS X 10.4; en-US; rv:1.9.0.7)
Gecko/2009021906 Firefox/3.0.7"
```

If we parse the single line of the server log into components, we will get the information like this:

**Table 1. Single line of server log parsed into components**

| | |
|---|---|
| Client IP Address | 71.70.185.34 |
| Date and Time Stamp | [09/Mar/2009:00:13:53 -0400] |
| Request URL | /search?Ntk=Keyword&Ne=2+200043+206475 +206590+11&N=206432&Ntt=boston+globe |
| Referring URL | http://search.lib.unc.edu/search?Nty=1&Ntk= Keyword&Ntt=boston+globe |

The URL parameters encode the user's search request. The parameters of interest are listed in table 2.

**Table 2. URL parameters**

| | |
|---|---|
| Search Field "Ntk" | Keyword |
| Text String (Query) "Ntt" | boston+globe |
| Facet Used "N" | 206432 (unique ID of facet value, stands for "Format: online") |
| Facet Opened to Browse "Ne" | 2+200043+206475+206590+11 ("Availability", "Location", "Format", "Subject", and "Publication Year") |
| 'Did you mean?' search feature on or off "Nty" | 1<br>The feature is on |

One of the difficulties for processing the log data is to identify individual users (sessions). For the log data, a single line of record which represents a request is treated as a transaction. Knowing only the IP addresses and query times, knowing how to chunk several transactions into a session is not trivial. Based on previous literature, we considered transactions occurring from the same IP address, and without a delay of 30 minutes or longer to be part of the same session. We identified 133951 search sessions for the dataset.

## 3.2 Coding Schema

An "action" refers to a user's interaction with the system. In most cases, a transaction represents a single action. Related to "actions" are the codes used to indicate generic categories of requests. There are a finite number of things that the user can manipulate when interacting with the system and therefore a finite number of codes representing actions. In this study, granularity is the main concern of adopting a coding schema. According to Wildemuth and Moore (1995), more detailed coding schemes are too fine-grained to make statistical analysis effective, while a coarser scheme would not provide enough detail. We decide to use both fine-grained coding and course-grained coding schemas to complement each other. One can determine the action the user took through transition from one state to another by comparing what has changed in the request URL and referring URL. In the example above, the only difference between the two URLs is the appearance of "Ne=2+200043+206475+206590+11&N=206432" in the request URL. Therefore, it is inferred that the user opened several facets and clicked one at this step.

There are totally 24 possible finer codes and 11 coarser codes. Since the amount of the log data is huge, all the coding work was completed automatically by Perl script by comparing the state change.

## 3.3 Cluster Analysis

To answer the second research question, cluster analysis is utilized. Cluster analysis is a statistical technique to create categories that fit observations. In this study, search sessions are to be clustered according to their similarity. In other words, the within-group sessions are alike while the between-group sessions are different. Relative proportions of action codes in a particular session are treated as the attributes for clustering. To reduce the computational cost and avoid subjectively predefining the number of clusters, a hybrid approach of hierarchical and non hierarchical methods is employed. In the first stage, 133951 sessions are divided into 100 clusters using a non-hierarchical method with the SAS procedure FASTCLUS. The output of this procedure serves as the input to the second stage, a hierarchical method using SAS procedure CLUSTER with Ward's algorithm. We adopted the coarser codes as the characteristic variables for the cluster analysis because the finer ones were too detailed to separate the clusters.

## 3.4 Sequential Event Analysis

To solve the third research question, sequential event analysis is applied to examine the action patterns of sessions. Specifically, two methods are employed. The first is Markov models checking transitions from one state to another. The other is maximal repeating patterns (MRP) identifying the actions sequences as long as possible for the searchers when they are searching the catalog.

### *3.4.1 Markov Models*

A Markov model is a stochastic process with the Markov property which means that the description of the present state fully captures all the information that could influence the future evolution of the process. Future states will be reached through a probabilistic process instead of a deterministic one. The order of Markov models means how many previous states (including the current state) influence the choice of the next state probabilistically. The simplest form of a Markov model is called a zero-order model. It is simply the frequency with which each state occurred. The first-order Markov model, also called a state transition matrix, reports the probability of the transition from all the possible current states to all the possible future states. First-order Markov models are the types of models most frequently found in the ILS literature (Wildemuth, 2009). Higher-order models can also be created and evaluated. A second-order Markov model takes into account the previous two states in trying to predict the next state, and so forth.

In this study, higher order Markov models are applied to describe the sequence of moves performed by searchers in a session. Furthermore, an order test (based on the chi-square goodness-of-fit, Anderson and Goodman (1963)) is conducted to indicate the statistical significance of whether the transitions are zero-, first-, second or higher order processes. For example, consider a session which could be represented as TB-MT-FM-FA-VR (the process of beginning a simple text search, then entering multiple terms in the search box, and then showing more values under one facet, next refining the search by adding a value of the facet, and finally viewing a record). If the process is tested to have the order of 2, significant action sequences of this session are all three-move thread, like TB-MT-FM. That is, the likelihood of being the state FM depends on having been in states TB and MT previously.

### *3.4.2 Maximal Repeating Pattern*

Previous literature indicates that people's information behavior varies greatly from one person to another. It is helpful to find patterns that are frequently adopted by searchers. Siochi and Ehrich's (1991) algorithm for identifying maximal repeating patterns (MRPs) among sequences of behavior is applied to serve this purpose. They defined an MRP as "a repeating pattern that is as long as possible, or is an independently occurring substring of a longer pattern" (Siochi & Ehrich, 1991, p.316). Thus, the algorithm systematically identifies those sequences of events that occur repeatedly within the data set. By examining what the MRPs look like and the frequencies at which they occur, we can pick out those we want to investigate further.

## 3.5 Visualization of Sequential Events

One of the primary goals of this study is to develop an automatic way to visualize the search action steps users take when searching the catalog. During the pilot study, a visualization method was developed for manually producing a graphical representation of action sequences within a session. Shapes and colors represent the different actions taken by the user. Twenty-five sample search sessions were chosen randomly and then graphical representations of their searchers were generated. Recreating a number of sessions manually helps the investigator better understand the coding issues, and the different search process, but is not feasible for analyzing millions of search records. Automated visualization tools or processes are required for this. Currently, work is underway to automatically generate XML that describes the actions, so that simple style sheets can be used to graphically render the coded actions for viewing using standard web browsers.

# 4. RESULTS

## 4.1 Are people really using facets to help them refine their search?

After processing through Perl scripts and MySQL database, the transactions were grouped into 133951 sessions. The average number of moves (actions) in a session is 9. The most frequently occurring moves are MultipleTermText and ViewRecord, which are the traditional operations in the classic library catalog. Facet Operations (OpenFacet, CloseFacet, ShowMoreFacet, AddFacet, RemoveFacet, RefineYears) only account up to 6% of the total moves. This number is much less than our previous study on the usage of NCSU's Endeca library catalog (40%; Cory, 2008) and still less than some other Endeca reports. It might be due to the fact that UNC has only been using this facet catalog for less than one year, and quite a few users are not familiar with it. Additionally, had we excluded some actions which are not of searching behavior kind, such as RSS Search and ViewRecord, the percent of facet operations would have increased.

## 4.2 Groups of Users

As mentioned above, a hybrid approach of non-hierarchical and hierarchical clustering techniques is employed to this study. As result, 8 clusters were identified based on the dendrogram and semi-partial R squared statistic. According to the characteristic variables (percentages of actions) of each cluster, we label these 8 groups as SimpleTextSearch group (few moves and most work is entering text string in the search box), DetailedTextSearch group (most work is evenly distributed in entering text string and viewing records), InDepthTextSearch group (clicking next pages frequently), AdvancedSearch group (using advanced search mode much more frequently), FacetTextSearch group (facet operations combined with text search), FollowupSearch group (clicking into the links provided by a particular record), RSS group (using RSS feed feature), and Outliers (conducted by Roberts).

## 4.3 Patterns of actions commonly adopted by searchers in FacetTextSearch group

Since FacetTextSearch (CL 8) is the population of interest in this research, Chi-square likelihood ratio test is employed to test the order of Markov model for this cluster. As result, the significant order is 2. That means the probability of being the current state depends on previous two states. Therefore, three-move sequence is the significant segment to describe the usage pattern in this group. The top 10 most frequent three-move sequences are summarized and the top pattern is all text search moves (TextSearch--TextSearch--TextSearch). Next most common is the pattern of all facet moves (ModifyFacets--ModifyFacets—ModifyFacets). The top two patterns together only account for 12% of all the possible patterns. We may infer that there is a wide distribution of usage patterns and that the common patterns among users are fewer than expected. From the 3rd to the 10th position, patterns are combination of text search, modifying facets and showing/hiding facets.

From the transition matrix (first order Markov model) generated for this search group, we know that the most likely preceding action for modifying facets is text search (38.61%) and for showing/hiding facets is also text search (36.59%). Therefore, we could infer that text search is assumed as the most likely starting point for faceted searches.

Applying the maximal repeating pattern algorithm (MRP) developed by Siochi to the facet search group, we identified 54 frequent (frequency higher than 500) patterns with 3 to 6 actions in each pattern. These 54 sequences are further grouped into three families: 1) facet search and then viewing record; 2) text search, facet search and then viewing record; 3) repeating (2) twice.

## 4.4 Visualizing Sequential Events

A set of rules was developed for the graphical representation of action sequence, with different shapes and flows representing the search process. For shapes: the green rectangle stands for TextSearch; dark green rectangle means AdvSearch; white rectangle means off catalog website; red rectangle denotes ViewRecord; yellow rectangle stands for NextPage; blue diamond means ModifyFacets; Bright blue diamond means

ShowHideFacets; white cloud means SortResults; and grey shapes means clicking the back button of the browser. In the chart, vertical flow stands for entering new search words which will generate a new result set, while the horizontal flow means refining the current search under the same text search words. Below is an example of the application of this rule to one search session. In this graph, the search words, the search field, the facet being incorporated, and the time spent on a particular manipulation are all displayed. The longer third and fourth lines indicate that the searcher kept refining his/her search through adding and removing facets or displaying facets.

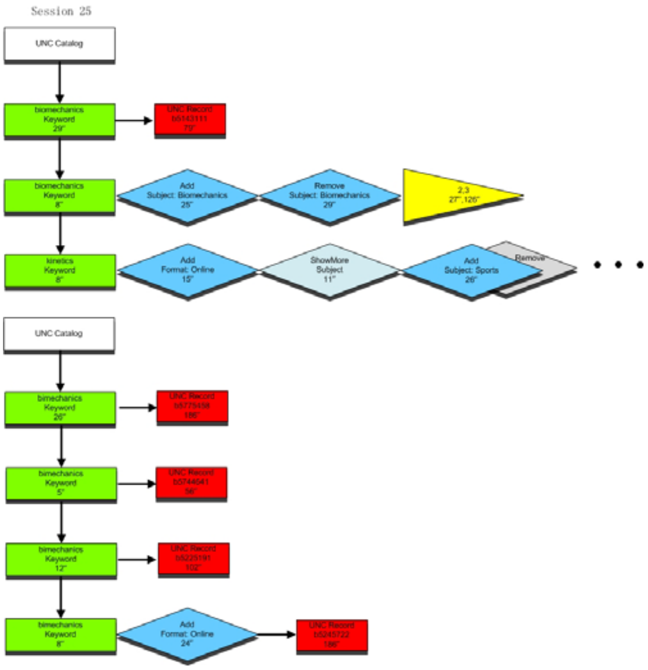


**Figure 1. Example visualization of one search session**

## REFERENCES

[1] Anderson, T., & Goodman, L. (1963). Statistical inference about Markov chains. Readings in Mathematical Psychology. (Vol. 2, pp. 241 – 262). New York: Wiley.

[2] Antelman, K., et al. (2006) Toward a Twenty-First Century Library Catalog.Information Technology and Libraries. 25(3) 128-139.

[3] Bates, M. J. (1979). Information search tactics. Journal of the American Society for Information Science, 30(4), 205-214.

[4] Borgman, C. (1996). Why Are Online Catalogs Still Hard to Use? Journal of the American Society for Information Science, 47(7), 493-503.

[5] Callender, J. (2001) Perl for Website Management. O'Reilly Media, Inc.

[6] Chapman, J. (1981). A state transition analysis of online information-seeking behavior. Journal of the American Society for Information Science, 32(5), 325-333.

[7] Chen, H. M., & Cooper, M. D. (2001). Using clustering techniques to detect usage patterns in a web-based information system. Journal of the American Society for Information Science and Technology, 52(11), 888-904.

[8] Chen, H. M., & Cooper, M. D. (2002). Stochastic modeling of usage patterns in a web-based information system. Journal of the American Society for Information Science and Technology, 53(7), 536-548.

[9] Goodrum, A. A., Bejune, M. M., & Siochi, A. C. (2003). A state transition analysis of image search patterns on the web. Lecture Notes in Computer Science, , 281-290.

[10] Jansen, B. J. (2005). Seeking and implementing automated assistance during the search process. Information Processing and Management, 41(4), 909-928.

[11] Jansen, B. J. (2006). Search Log Analysis: What Is It; What's Been Done; How to Do It. Library and Information Science Research, 28(3), 407-432.

[12] Jansen, B. J., Spink, A., Blakely, C., & Koshman, S. (2007). Defining a Session On Web Search Engines. Journal of the American Society for Information Science and Technology, 58(6), 862-871.

[13] Koch, T., Golub, K., & Ardo, A. (2006). Users browsing behaviour in a DDC-based web service: A log analysis. Cataloging & Classification Quarterly, 42(3-4), 163-186.

[14] Lown, C. (2008). A transaction log analysis of NCSU's faceted navigation OPAC.

[15] Novotny, E. (2004). I Don't Think, I Click: A Protocol Analysis Study of Use of a Library Online Catalog in the Internet Age. College & Research Libraries, 65(6),525–37.

[16] Olson, T.A. (2007). Utility of a faceted catalog for scholarly research. Library Hi Tech, 25(4 ), 550-561.

[17] Qiu, L. (1993). Markov models of search state patterns in a hypertext information retrieval system. Journal of the American Society for Information Science, 44(7)

[18] Siochi, A.C., & Ehrich, R.W. (1991). Computer analysis of user interfaces based on repetition in transcripts of user sessions. ACM Transaction on Information Systems, 9(4), 309 – 335.

[19] Spink, A. (1996). Multiple search sessions model of end-user behavior: An exploratory study. Journal of the American Society for Information Science, 47(8)

[20] Wildemuth, B. M. (2004). The effects of domain knowledge on search tactic formulation. Journal of the American Society for Information Science and Technology, 55(3), 246-258.

[21] Wildemuth, B.M., & Morre, M.E. (1995). End-user search behaviors and their relationship to search effectiveness. Bulletin of the Medical Library Association, 83(3): 294 – 304.

[22] Yu, H. & Young, M. (2004) The Impact of Web Search Engines on Subject Searching in OPAC. Information Technology and Libraries. 23(4), 168-180.

# Freebase Cubed: Text-based Collection Queries for Large, Richly Interconnected Data Sets

David F. Huynh
Metaweb Technologies, Inc.
631 Howard Street, Suite 400 San Francisco, CA 94105
david@metaweb.com

## ABSTRACT

Any large data set such as Freebase that contains a large number of types and properties accumulated over actual use rather than fixed at design time poses challenges to designing easy-to-use faceted browsers. This is because the faceted browser cannot be tuned with domain knowledge at design time, but must operate in a generic manner, and thus become unwieldy.

In this work, we propose that support for a particular kind of text-based queries can let users perform faceted browsing and set-based browsing operations on such data sets with the ease and familiarity of conventional keyword search. For example, the text query “german car companies founders” can replace the actions of filtering all Freebase data by type to “company”, by industry to “car”, and by country to “Germany”, and then pivoting to those companies’ founders. From there, the user can perform faceted browsing actions to refine the already narrow collection further. We describe an algorithm for parsing these *collection queries* and demonstrate an implementation that works on Freebase.

## Categories and Subject Descriptors

H.5.2 [**User Interface**]: Interaction Styles, Natural Language.

## General Terms

Algorithms, Design, Human Factors, Languages.

## Keywords

Search, pidgin, faceted browsing, set-based browsing, graph data.

## 1. INTRODUCTION

The faceted browsing paradigm has been very effective in letting casual users browse through large data sets by performing simple actions of picking suggested filters to apply. This paradigm remains effective as long as the schemas in the system are known a priori so that the interface can be configured based on the schemas. For example, an online retailer can be expected to know all types of product that it offers, as well as important aspects of each type of product (e.g., resolution for televisions, maximum zoom for cameras). Such domain knowledge helps configure the faceted browser, making it optimal for the domain in question.

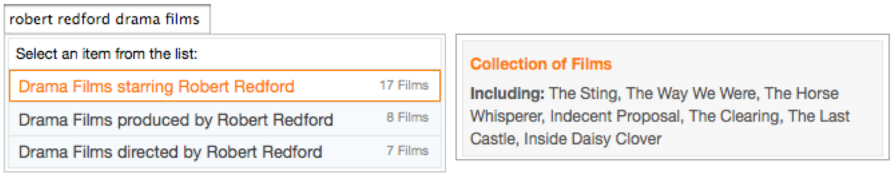
Domain knowledge is difficult to obtain and use in such a data set as Freebase in which new schemas are added over actual use rather than fixed at design time. Freebase currently contains some 3,000 types (e.g., company, book author) and over 30,000 properties (e.g., country where a company is founded, books written by an author). Users can add new topics to existing types or they can add entirely new types and properties. Providing a faceted browser over even just the stable types and properties is still a challenge for two reasons.

- Consumer-facing faceted browsers typically have one over-arching type under which all data can be organized. For example, online retailers deal primarily with products; libraries deal with books. In Freebase, there is no over-arching type: users are as likely to search for companies as they are for book authors, or any one of the 3,000 types.
- Types in existing faceted browsers are isolated. When users search for televisions in an online retailer, there’s little chance that they would be concerned with cameras at the same time. In contrast, on Freebase where types are highly interconnected, users might want collections defined by multiple types, such as “pharmaceutical companies funding republican politicians’ campaigns.” The more interconnected the types are, the more the potential ways to define collections, and the more facets the browser has to offer.

These challenges are inherent in any data set that resembles Freebase. That includes other comprehensive semantic web data sets such as Dbpedia, or even smaller, personal semantic web data sets accumulated by gathering tidbits from several data sources using something like Tabulator [3] or Piggy Bank [4].

We note that these challenges are acute at the beginning of any faceted browsing session when the collection to deal with is still large and/or heterogeneous. After applying a few filters, the collection gets small and homogeneous enough for faceted browsing to be effective again.

We propose that just when the user wants to search a large, interconnected data set, a particular class of text-based queries can be supported for filtering the data down to a manageable



robert redford drama films

Select an item from the list:

Drama Films starring Robert Redford 17 Films
Drama Films produced by Robert Redford 8 Films
Drama Films directed by Robert Redford 7 Films

Collection of Films

**Including:** The Sting, The Way We Were, The Horse Whisperer, Indecent Proposal, The Clearing, The Last Castle, Inside Daisy Clover

**Figure 1. As the user types a query into the Freebase Cubed suggest widget, the query is interpreted and the interpretations are shown in a drop-down menu. When an interpretation is hovered using the mouse or selected using the keyboard, a fly-out appears to show the interpretation’s details.**

collection. These *collection queries* have a syntax that maps to faceted browsing operations (filtering) and set-based browsing operations (pivoting), and are sufficiently natural for casual users. An example is "german car companies founders", which maps to filtering by type to company, by industry to car, by country to German, and then pivoting from those companies to their founders. Other examples include:

- frank wright buildings
- female african american politicians
- kate winslet drama films directors

We observe that some existing web searches already follow this syntax. Issuing such a query today performs a keyword text search, whereas we propose that it be translated to and performed as a structured query.

We describe an algorithm for parsing collection queries, demonstrate a prototype called "Freebase Cubed," and show how, as a simple search textbox, it fits into more places than faceted browsing UIs can (Figure 1). We plan to use this widget on Freebase.com and in Freebase-powered web applications to introduce users early to collections and surface the richness of the data within Freebase even in places where full-blown faceted browsing UIs cannot be afforded.

## 2. GENERAL DEMAND

While Freebase as an enormous, richly-interconnected data source needs something more than the conventional faceted browsing paradigm to be accessible to the general public, we wanted to further determine if that solution will also benefit the Web itself, and the Data Web to come. To know if web users already issue web searches resembling collection queries, we conducted a preliminary investigation over web searches relevant to the types in Freebase using Google Insights [1] and Google Suggest API [2]. For each of the few hundred most populated types in Freebase, we pluralized its name (e.g., /government/polician to "policians") and submitted that text to Google Insights, from which we could download a table of web searches that Google deemed related to that text (e.g., "indian politicians", "female politicians"). For each of those related web searches, we ran it through the Google Suggest API to retrieve its search volume. While there is no official documentation for the Google Suggest API, developers on the Web have assumed that the "num_queries" values that it returns are search volumes. For example, Google Suggest API returns 102 million "num_queries" for "black politicians". If this assumption holds, our investigation indicates sizeable search volumes for web searches related to type names in Freebase, and many of these web searches do take the form of collection queries.

From this investigation, we contend that these non-negligible search volumes on what we consider to be collection queries are evidence that some web users already want to search for information on collections (e.g., "black politicians") rather than single, specific topics (e.g., "barack obama"). Not only does this early result signify demand, but it also suggests that if we were able to support collection queries, then users would be ready to use them without much learning.

## 3. INTEGRATION USE CASES

In addition to the need to make Freebase browse-able and the general demand for supporting collection queries, we also have another family of use cases in mind. Freebase is a rich source of data that can be used to augment web sites. For example, a prolific political blogger might wish that her readers can easily browse through or search her blog posts not just by the names of politicians mentioned, but also by the politicians' attributes, such as their parties, their religions, their states, etc., which might or might not be mentioned explicitly in each post. All such data is or can be maintained in Freebase, and if each blog post carries Freebase identifiers of politicians, or any other topic that it also mentions, then compelling search and browse interfaces can be built on top of the two data sources: the blog's index of topic identifiers for each post, and Freebase.

Depending on each particular integration scenario, there might or might not be enough screen real estate for a full-blown faceted browser. There might be just enough room for a search text field, but a desire to support semantically rich searches nevertheless. Even if there is room for a faceted browser, configuring the browser any way would not leverage all the data in Freebase. So while the blogger might anticipate the need for facets such as "political party," "state," and "religion," the reader looking for "movie actor politicians" won't be satisfied by those facets. A more free-form mechanism is needed, and supporting text-based collection queries is one possible answer.

Supporting collection queries in a particular embedding scenario has different requirements than on the Freebase site. Specifically, we want to support shortcuts such that, say, on a book review blog, the user can just type "african american authors" rather than "african american authors' books" to search for books by African American authors.

## 4. ALGORITHM

A text query might be targeting a single topic or a collection of topics. Thus, given a text query, we would need to make either single topic suggestions or collection suggestions, or both. In order to make collection suggestions, we need to interpret the query as a collection query; this is discussed in 4.1. In order to suggest some combination of single topic suggestions and collection suggestions, we need an overarching algorithm for generating such combination; this is discussed in 4.2.

The entire discussion is posed in the context of Freebase, but it should be applicable to any similar heterogeneous graph data set. The natural language in which text queries are posed is assumed to be English. The generic prototype called Freebase Cubed is accessible at http://cubed.freebaseapps.com/, and a book embedding demo is available at http://cubed.freebaseapps.com/embed-books.

### 4.1 Collection Query Interpretation

A text query is a collection query when, by our definition, it can be translated to a sequence of filtering and/or pivoting operations. Assuming that a text query is a collection query, we interpret it in three phases.

#### *4.1.1 Chunking*

First, we decide how the text query should be broken down (chunked). For example, "robert redford drama films" can be chunked in many ways, e.g.,

- robert + redford drama films

- robert redford + drama films
- robert redford drama + films
- robert + redford + drama films
- robert redford + drama + films

In each chunking solution, each chunk is supposed to correspond to a topic (such as /en/robert_redford), or a type (e.g., /film/film), or a property (e.g., /film/film/directed_by). To determine what each chunk matches, we query for the chunk's text against Freebase's text search service. Several search matches are kept per chunk, but the best match dictates the chunk's *form* (type, property, or topic). Any chunk whose text is a plural noun (e.g., "films" rather than "film") is biased to type or property form more than topic form. Demonyms (e.g., Canadian) are resolved to their corresponding countries (e.g., /en/canada).

Chunking solutions can be ordered by how well the chunks in each solution match against data and schema in Freebase. They are fed in that order into the next phase.

### *4.1.2 Seeding*

Given a chunking solution which consists of an ordered list of chunks, this phase picks one of those chunks to be the starting point from which filtering and pivoting operations are applied. There are two types of seed:

- seed collection defined by a type chunk, such as "authors" in "french authors' books"
- seed topic defined by a topic chunk, such as "jfk" in "jfk's children".

As there are substantially fewer types than topics, a type match in the chunking phase is much less ambiguous than a topic match. Thus, as a rule of thumb in picking seeds, we favor type-form chunks over topic-form chunks.

A chunking solution plus the choice of a seed form a seeding solution. A structured query called the seed query is formulated to represent the seed collection or the seed topic.

### *4.1.3 Growing*

Given a seeding solution, this next phase interprets the rest of the chunks, one by one, as filtering and pivoting operations. For example, having chunked "french authors' books" into french + authors + books and picked the "authors" chunk as the seed collection, this phase interprets the "french" chunk as a filtering operation (filtering by nationality), and the "books" chunk as a pivoting operation (pivoting from authors to their works).

The chunk immediately to the left of the already interpreted chunks is considered first, and then the chunk to the right, in order to favor the adjective-noun ordering of English. For example, starting from the seed "films" in the collection query "drama films actors," we consider filtering the films by genre first before considering to pivot to the films' actors. If the user phases the query as "films drama actors," we can still interpret it.

Type-form chunks and property-form chunks are interpreted as pivoting operations, and topic chunks are interpreted as filtering operations. Applying such an operation means extending the current structured query either by adding a constraint for filtering or by wrapping the current query as a nested query for pivoting.

The current structured query denotes a single topic of some types or a collection of topics sharing some common types. The chunk to be considered for growing that query is also associated with one or more types. For example, in the query "drama films actors," the chunk "drama" is typed /film/genre, and the chunk "actors" is typed /film/actor. The seed collection "films" is typed /film/film. To grow the seed collection with the chunk "drama", we retrieve properties that point from type /film/film to /film/genre. There is only one such property: /film/film/genre. Next, to grow the collection of drama films with the chunk "actors", we retrieve properties that point from type /film/film to /film/actor, and we get /film/film/starring.

There are cases where we get more than one connecting property. Consider the query "robert redford films" in which "robert redford" is typed /film/actor, /film/director, and /film/producer. In growing the seed collection "films" with the chunk "robert redford", we get three different connecting properties: /film/film/starring, /film/film/directed_by, and /film/film/produced_by. These lead to three final interpretations of the query which are shown as three suggestions (Figure 1); and the user is asked to select one.

Note that for each chunk, the chunking phase keeps several search matches. These are useful when the chunk's best match depends on other chunks in the text query. For example, in the query "apple products", the universal best match of "apple" is the fruit, but in the context of "products", the best match is Apple Inc. the company.

While we typically grow the structured query one chunk at a time, when we encounter a property-form chunk, we can use it to qualify the next immediate chunk, and in doing so, grow the query by two chunks in one shot. For example, in the query "robert redford directed films," the chunk "directed" matches the property /film/film/directed_by, which we use to qualify the connection between "films" and "robert redford". This leads to a single interpretation (rather than three previously).

Given a seeding solution, when all chunks have been used up in growing the seed, we have one possible complete *interpretation* of the original text query, which can be expressed as a structured query. A single seeding solution can yield several interpretations.

### *4.1.4 Decision Tree and Greedy Implementation*

All three phases—chunking, seeding, and growing—involve many decisions to make, each of which has many possible solutions. The whole process can be viewed as a decision tree in which the leaves are the complete interpretations of the original text query at the root of the tree. In our prototypical implementation, this decision tree is traversed depth-first, and at each node, branches are ordered by local scores, yielding fast but greedy performance.

### *4.1.5 Pseudo-types*

Implicit in our discussion so far is the assumption that a user's notion of a general collection of topics (e.g., "films") is modeled as a type in Freebase (/film/film). This is not always true. For example, we might expect "volcano" to correspond to a type in Freebase, but it does not. Rather, it corresponds to a kind of mountain, and "mountain" corresponds to the type /geography/mountain. Thus, "volcanoes" is translated to a structured query for topics of type /geography/mountain and having /geography/mountain/mountain_type /en/volcano. This is our concept of *pseudo-types*, which bridge the gap between the user's notion of types and the actual types in Freebase.

## 4.2 Unified Suggest Widget

The previous sub-section discusses the core algorithm for interpreting a collection query. In real use, the user might enter a single topic query, or, as discussed in section 3, might enter an abbreviated collection query. In order to accommodate as all three kinds of query, we need an overarching algorithm, which consists of many more phases as discussed below.

### *4.2.1 Unifying*

Given a text query, we submit it as-is to the Freebase search service and retain some top results based on some threshold criteria. Next, we try to match the query against past collection queries that other users have issued. For example, when the user just types "french," we can get the partial matches "french authors," "french wines," etc. These partial matches both save the user from typing as well as hint the user of our novel collection query support. If all of these matches are partial, then we interpret the query using the core algorithm discussed in 4.1. The output of this phase is a list of zero or more single topic search matches, zero or more partial matches of past collection queries, and zero or more interpretations of the text query as a collection query.

### *4.2.2 Extending*

To support collection query in an embedding scenario within a specific context, as discussed in section 3, we also need to understand abbreviated queries. The suggest widget can be configured to give hints about which types to expect from a query, and if it is abbreviated, how to generate a full query from that. For example, in a book embedding scenario, full queries should be of type /book/written_work, and abbreviated queries can be of type /book/author or /book/literary_genre. Knowing these types helps us quickly find appropriate partial matches from past queries in the unifying phase.

### *4.2.3 Condensing and Approving*

This phase eliminates duplicate interpretations from the previous phase as well as interpretations that resolve to empty collections (due to lack of data in Freebase or in the real world).

### *4.2.4 Explaining*

This last phase generates natural language text explaining each interpretation back to the user. For example, the text query "robert redford films" can be interpreted in three different ways, and explained back to the user as "films starring Robert Redford," "films directed by Robert Redford," and "films produced by Robert Redford." The algorithm for generating a textual explanation from the structured query of an interpretation is complicated and not yet sufficiently fleshed out to be explained here.

# 5. RELATED WORK

Kaufmann's doctoral thesis [5], which investigates natural language interfaces to semantic web data, is a highly related body of work. One of the systems she built, NLP-Reduce, translates text-based natural language queries into structured queries also using a pattern-matching approach like ours. NLP-Reduce is even more forgiving in that it removes even more stop words and allows for full questions or fuller question fragments. But whereas NLP-Reduce is aimed to address generic questions (e.g., "how big are the lakes in Illinois?"), our work aims to only retrieve collections of topics. Our focus allows us to make assumptions about shortcuts that users would tend to make, particularly that they would phrase queries as lists of keywords that map to filtering and pivoting operations. We believe that this is closer to how web users use existing search engines. Furthermore, we are unable to verify how well NLP-Reduce would work on a large and heterogeneous data set as Freebase. Kaufmann's systems have only been evaluated on 3 data sets, each having no more than 10 types, 20 properties, and 10,000 topics (instances). On the other hand, Freebase has 3,000 types, 30,000 properties, and almost 9 million topics. This difference in magnitude should have implications on both data processing performance as well as the effectiveness of heuristics: large, heterogeneous data sets tend to have more name collisions, and the more flexible the query is allowed to be, the more explosive the number of interpretations there are.

# 6. FUTURE WORK

While we contend in section 2 that some web users already formulate collection queries, it is not clear how they would react to precise collections as search results as opposed to million fuzzy keyword matches that existing search engines return. It is also not clear if suggestions from partial matches against past collection queries are enough to make web users aware of this new capability and confident to formulate new, similar collection queries themselves. If they are enough, then we can prime the past collection query index with pre-canned queries generated from some typical patterns such as "<nationality> <profession>". Finally, the Freebase Cubed suggest widget requires some UI iterations and usability testing to make sure that users understand the difference between single topic suggestions and collection suggestions. Those are some of the research tasks to be done next.

# 7. REFERENCES

[1] Google Insights. http://www.google.com/insights/search/.

[2] Google Suggest API.

[3] Berners-Lee, T., Y. Chen, L. Chilton, D. Connolly, R. Dhanaraj, J. Hollenbach, A. Lerer, and D. Sheets. Tabulator: Exploring and Analyzing Linked Data on the Semantic Web. ISWC, 2006.

[4] Huynh, D., S. Mazzocchi, and D. Karger. Piggy Bank: Experiencing the Semantic Web inside Your Web Browser. ISWC 2005.

[5] Kaufmann, E. Talking to the Semantic Web? Natural Language Query Interfaces for Casual End-users. Doctoral thesis, 2008.

# System Controlled Assistance for Improving Search Performance

**Bernard J. Jansen**
College of Information Sciences and Technology
The Pennsylvania State University
University Park, Pennsylvania 16802
jjansen@acm.org

## ABSTRACT

This position paper outlines the concept of system assistance as a method to improve searching performance. I present an investigation concerning the effects of user-controlled versus system-controlled assistance on searching performance using a within subjects, counterbalanced empirical evaluation. Forty-three subjects interacted with two fully functional, information retrieval systems offering searching assistance based on implicit feedback. The systems were identical in all respects except that one offered searching assistance via a help link, and the other offered system-controlled support at specified points during the search progress based on patterns of searcher interactions. The evaluation used the W2G Text REtrieval Conference document collection with six topics. Research results indicate that offering system-controlled assistance based on patterns of implicit feedback can improve searching performance based on user selected relevant documents, with an approximately 30% performance increase overall. I discuss the implications for the design of future searching systems with assistance that is based on user implicit feedback patterns.

## Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*query formulation*, *search process*; H.3.4 [**Information Storage and Retrieval**]: Systems and software—*performance evaluation* (*efficiency and effectiveness*)

## General Terms

Design, Experimentation, Human Factors, Verification.

## Keywords

Searching assistance, explanations, implicit feedback, personalization, searching systems, user evaluation

## 1. INTRODUCTION

The improvement of searching systems is an active research area, with the aim of addressing some of the issues users have when interacting with these systems [6, 8]. Searcher issues include finding appropriate query terms, retrieving too many results, not retrieving enough results, and retrieving zero results [8], among many others. These issues are especially pronounced in searching contexts where users lack the domain knowledge or contextual awareness to use effectively the searching system. These types of searches are typified by user uncertainty about the information need, the content space, or the search engine's capabilities.

To address these situations, contextual help and similar information retrieval (IR) systems attempt to aid the searcher during the search process by either executing search tactics for or offering assistance to the user in order to help in locating relevant information. These systems usually rely on implicit feedback. Ingwersen and Belkin [2] highlight that IR systems will increasingly depend on implicit interactions in order to improve IR performance.

However, there has been little empirical evaluation of whether or not this searching assistance is beneficial to users during the search process. Is this assistance helpful? If so, when is helpful? When do searchers desire assistance? The research results presented in this article address these questions. We specifically examine whether system-controlled searching assistance is beneficial to users during the search process.

The paper begins with a short review of literature concerning IR systems offering searching assistance. I then provide a description of the two applications we developed and utilized in this research. Next, I discuss the empirical study we conducted to evaluate the effectiveness of system directed assistance on searching performance. The paper then presents the results of an analysis, draws implications for searching system design, and then discusses directions for future research.

## 2. RESEARCH QUESTIONS

This research is a user study utilizing a searching assistance application developed to investigate some of these issues. The research question is *Does system-controlled searching assistance based on patterns of implicit feedback improve performance during the search process?*

The research hypothesis: *There is a significant increase in searching performance when using system-controlled searching assistance compared to a system offering user-controlled assistance during the search process, as measured by the number of relevant documents that the user selects during a session.*

I measure the number of documents that the user selects as relevant from all documents retrieved in a session. A session is defined as: *an episode of a searcher using a searching system during which a series of interactions occur between the searcher and system*. I considered a document relevant based on implicit feedback from the user and explicit qualitative data indicating relevance. For example, during the experiment, a searcher may bookmark a document and state that the document was relevant to the information need.

# 3. SYSTEM DEVELOPMENT

For this research, I used a client-side software application developed, possessing a suite of searching assistance features. Our application development goal was that the component would rely on implicit feedback, gathering information solely from normal user – system interactions occurring during the search process.

The paper presents a brief overview of the system (shown in Figure 1), with two earlier versions of the application presented in [3, 4]. These earlier applications were solely user-controlled. User evaluations of these applications pointed to the need for more system control of assistance during the searching process, resulting in the research presented here.

## 3.1 System Design

When a session begins, the application monitors the searcher via a wrapper to the browser. When the application detects a valid action, it records that action and the specific object receiving the action. For example, if a searcher was viewing a Web page and bookmarked it, the application would record this as (*bookmark Web page*). The application then associates appropriate search assistance for the user based on the particular action and the system's analysis of the object. In this example, the system would offer the searcher relevance feedback terms from *Web page*. The more the system records and integrates these *(a, o)* pairs, the more complex could be the model of the information need. The application currently monitors the searcher's interactions with the system, tracking implicit feedback actions. The application logs actions of *bookmark, copy, email, print, scroll, save, execute* (i.e., *submit and click)* and *view* (*without scrolling*).

## 3.2 Automated Assistance

The application monitors the session for one of the implicit feedback actions and associated objects using a browser wrapper. When the system detects a valid action (i.e., (*a, o*) pair), it records the action and the specified object receiving the action. The system then offers appropriate search assistance to the user based on the particular action, the pattern of previous interaction during the session (for the system-controlled version of the application), and the system's analysis of the object. The current searching assistance features of the application are:

1. **Managing Results:** The automated assistance application provides suggestions to improve the query in order to either increase or decrease the number of results using query operators.

2. **Query Refinement:** The system uses the Microsoft Office thesaurus to suggest other query terms, but the application can utilize any online thesaurus via an application program interface.

3. **Query Reformulation**: The system displays similar queries from prior users based on number of previous submissions and terms in the current query.

4. **Relevance Feedback**: The system implements a version of relevance feedback using terms from a user selected document or passage object. The system provides suggested terms from the document that the user may want to implement in a follow-on query.

5. **Spelling:** The system offers spelling suggestion using the system's online dictionary, Microsoft Office Dictionary, although it can access any online dictionary via the appropriate API.

For the user study, there were two versions of the system. In the user-controlled version of the system, the *Assistance* module displayed searching assistance whenever it was available via a "View Help" button. In version two of the system, the *Assistance* module displayed assistance only when the *Pattern Recognition* module detected pre-set patterns.

## 3.3 Pattern Recognition

The *Pattern Recognition* module (see Figure 1) was based on prior work [3], where we conducted exploratory sequential data analysis of users interacting with searching. The analysis was conducted to determine when in the search process users desire system intervention. Using transaction logs, videotapes, and lab notes from this study [3], I coded the user – system interactions for each subject. Once I had coded all interactions, I sequentially ordered these interactions (i.e., states) for each searcher.

From these findings, a module was developed to monitor patterns and interject assistance only at certain points in the search process in order to reduce task interruption. This *Pattern Recognition* module accepts implicit feedback data from the *Tracking* module, storing *(a, o*) pairs and implicit feedback actions.

When the Pattern *Recognition* module identifies pre-coded implicit feedback patterns, it passes the *(a, o*) pairs to the appropriate module and alerts the *Assistance* module to display the assistance. The current preset patterns and are:

- *Execute Query – View Results Page (with scrolling or without scrolling).*
- *Implicit Relevance action (i.e., bookmark, copy, print, save) – Navigation (of the browser).*
- *Implement automated assistance – View Results Page.*

If the Pattern module does not detect these conditions, then the system does not display any searching assistance. The image is of the assistance that is automatically displayed when the application detects a set pattern.

# 4. USER STUDY

We used two systems in this evaluation that were identical in all respects, except that one offers system-controlled searching assistance, and the other user-controlled assistance. The backend searching systems used for the empirical study were Microsoft's Internet Information Service (IIS). The IIS systems were running on an IBM-compatible platform using the Windows XP operating system and Microsoft Internet Explorer as the systems' interfaces.

For the system-controlled searching assistance, we integrated the assistance application via a wrapper to the Internet Explorer browser. For the user-controlled system, we used a duplicate automated assistance application with the *Pattern Recognition* module disabled so that the system would display the searching assistance in the browser whenever available.

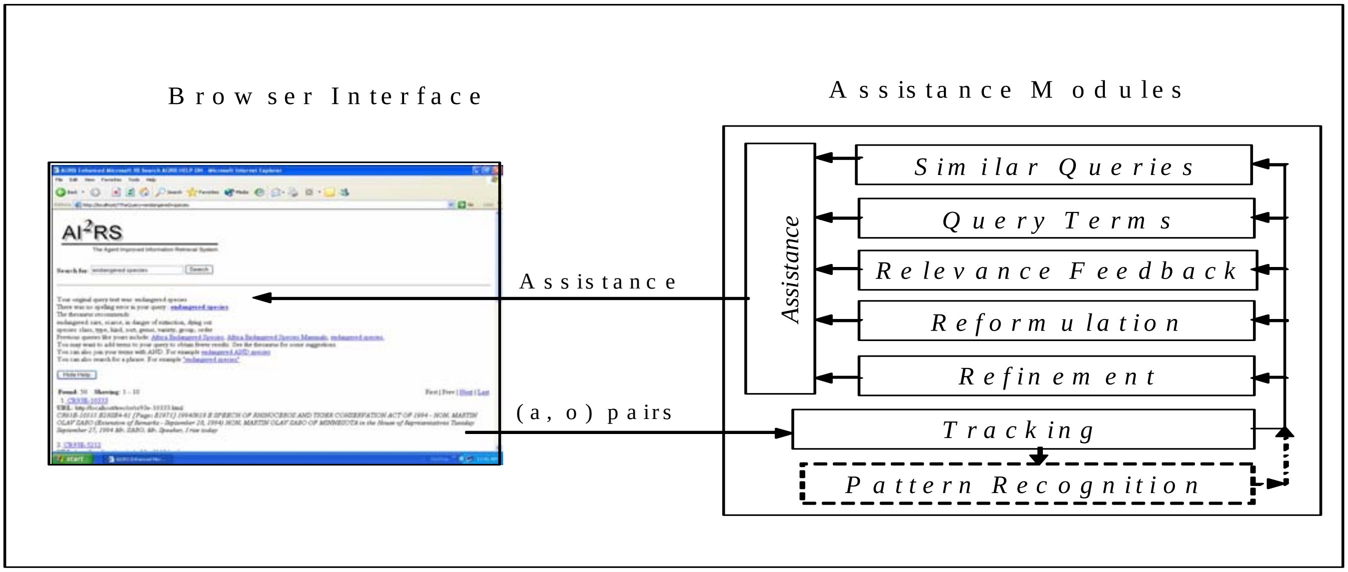


**Figure 1. Searching Assistance Modules and Information Flow with the Browser Interface.**

## 4.1 Pre-Study Measures

The subjects for the evaluation were 43 college students attending a major U.S. university. All were familiar with the use of Web search engines. Most of the students were studying information science and technology or other aspects of engineering. So, our sample has an understanding of computers and information technology. The subjects were given no additional training on the searching systems, a pre-evaluation demographic survey was administered, used previously [3].

## 4.2 Document Collection and Topics

The study used the W2G Text REtrieval Conference (TREC) document collection with six topics. We parsed the aggregate files into their individual component documents. The TREC collection is a standard test collection for online search systems (see http://trec.nist.gov/ for more information). The test collection after parsing contained approximately 200,000 documents. Each TREC collection comes with a set of topics for which there are relevant documents in the collection. The six topics we used for this study were: *Behavioral genetics*, *Tropical Storms*, *Quilts being used to generate income*, *Robotic technology*, *Estonia economic issues*, and *Super critical fluids*. In the results reported here, we were interested in the documents that the users selected as relevant, so we did not utilized the TREC relevant judgments.

## 4.3 Experimental Set-up

At the start of the study, I provided each of the subjects a short statement instructing them to search on a given topic in order to prepare a report, which is in line with the definition of relevance judgments for the TREC documents. The subjects had fifteen minutes on each system to find as many relevant documents as possible. We determined the length of the search session based on reported measures of the typical Web search session [5].

The subjects were notified that the systems contained an automatic feature to assist them while they were searching. When the subjects utilized the user-controlled system, they were shown a screen capture of the assistance button and an example of the assistance shown if they clicked the button. They were instructed that they could access the assistance by clicking the button, or they could ignore the offer of assistance with no detrimental effect on the system. When the subject used the system-controlled assistance, we showed them a screen capture of an example of the displayed assistance. Again, the subjects were instructed that they could view the assistance or ignore the assistance with no detrimental effect on the system.

For the searching sessions, each of the subjects was given one of the six search topics, read the one paragraph explanation provided with the TREC collection, and then afforded the written explanation to them. They were asked to search as when they normally conduct online research, taking whatever actions they usually take when locating documents of interest online. In this respect, I adhered to recommendations to place the searching need within a scenario [1, 7].

All subjects used both systems, searching on a different topic on each system. The systems were counterbalanced to ensure that an equal number of subjects searched first on each system. The topics were also rotated after every sixth subject to ensure topic order did not introduce learning effects that would bias the searching performance.

## 5. RESULTS

The following sections present the results of the empirical evaluation. The hypotheses were evaluated by performing a paired t-test using the number of relevant documents identified by the study participants during their sessions on each system. There was a significant difference in performance between the two systems ($t = 2.553$, $p < 0.01$, $df=42$). Therefore, we fail to reject the hypothesis; there is a statistically significant performance improvement with a system-controlled assistance searching system. Table 1 displays the number of relevance documents identified by participants using the user-controlled assistance and the system-controlled assistance.

There were thirty subjects (70%) who located more relevant documents using system-controlled assistance compared to 9 (21%) who preformed better on the system with user-controlled assistance. Four subjects (9%) located the same number of relevant documents on both systems.

**Table 1. Identification of Relevant Documents.**

| | **User-Controlled** | **System-Controlled** |
|---|---|---|
| Relevant Documents From All Users | 175 | 227 |
| Mean Number of Relevant Documents | 4.07 | 5.28 |
| Standard Deviation | 2.87 | 3.72 |
| | | |
| Subjects Locating More Relevant Documents on System with Automated Assistance | 30 | 70% |
| Subjects Locating More Relevant Documents on System without Automated Assistance | 9 | 21% |
| Subjects Locating Same Number of Relevant Documents on Both Systems | 4 | 9% |
| Total | 43 | 100% |

## 6. DISCUSSION

In 70% of the cases (30 subjects), searchers on the system with system-controlled searching assistance performed better than on the system user-controlled searching assistance. This is especially noteworthy since the assistance was based totally on implicit feedback, which is not as exact as explicit feedback. However, the Web is a natural environment for the use of implicit methods, and our results indicate that it is a worthwhile area to pursue.

However, there were also 30% of the users that were not helped by or performed worse on the system-controlled searching assistance system. There were 9 searchers who performed worse, and 4 searchers who performed the same on both systems. This would indicate that one might not be able to apply system-controlled assistance techniques wholesale and still achieve maximum outcome. Rather, a more individualized and targeted approach within the context of the searching process may be worthwhile.

This also indicates that individual differences are also likely active when it comes to utilizing and accepting searching assistance to improve performance, as is the case for other human-computer interface applications.

The limitations of the study are the use of the TREC topics and scenarios, which may not reflect the difficulty level of many searching tasks. However, these tasks are certainly reflective of many challenging information needs such as exploratory searching and competitive intelligence. Another limitation is the requirement of the participants to locate as many relevant documents as possible, which is not reflective of the retrieval goal of other searching tasks such as home page or fact finding. However, this is a common searching goal for reports, market research, health issue, or other tasks where the searcher wants to become informed on a topic.

## 7. CONCLUSION

The results of the research conducted so far are very promising. In this paper, we present the design of a general purpose system-controlled searching assistance application that uses implicit feedback to provide recommendations on searching tactics. We evaluated the use of user-controlled versus system-controlled assistance with real users in order to measure the performance benefit of system-controlled searching assistance. This assistance was based on patterns of implicit feedback.

## 8. REFERENCES

[1] Borlund, P. and Ingwersen, P. The development of a method for the evaluation of interactive information retrieval systems. *Journal of Documentation*, *53* (5). 225-250.

[2] Ingwersen, P. and Belkin, N. Information Retrieval in Context - IRiX. *SIGIR Forum*, *38* (2). 50-52.

[3] Jansen, B.J. and McNeese, M.D. Evaluating the Effectiveness of and Patterns of Interactions with Automated Searching Assistance. *Journal of the American Society for Information Science and Technology*.

[4] Jansen, B.J. and Pooch, U. Assisting the Searcher: Utilizing Software Agents for Web Search Systems. *Internet Research - Electronic Networking Applications and Policy*, *14* (1). 19-33.

[5] Jansen, B.J. and Spink, A., An Analysis of Web Information Seeking and Use: Documents Retrieved Versus Documents Viewed. in *4th International Conference on Internet Computing*, (Las Vegas, Nevada, 2003), 65-69.

[6] Jansen, B.J., Spink, A. and Saracevic, T., Failure Analysis in Query Construction: Data and Analysis from a Large Sample of Web Queries. in *3rd ACM Conference on Digital Libraries*, (Pittsburgh, PA, 1998), 289-290.

[7] Rosson, M.B. and Carroll, J.M. *Usability Engineering: Scenario-Based Development of Human-Computer Interaction*. Morgan Kaufmann, New York, 2002.

[8] Yee, M. System Design and Cataloging Meet the User: User Interfaces to Online Public Access Catalogs. *Journal of the American Society for Information Science*, *42* (2). 78-98.

# Designing for Enterprise Search in a Global Organization

Maria Johansson
Findwise AB
Drottninggatan 5
411 14 Gothenburg, Sweden
maria.johansson@findwise.se

Lina Westerling
Findwise AB
Sveavägen 31
111 34 Stockholm, Sweden
lina.westerling@findwise.se

## ABSTRACT

Enterprise Search is used by organizations to capitalize on their internal knowledge by providing quick access to all internal information, helping users re-finding and discovering new information, as well as creating the necessary conditions for collaboration across organizational and geographical boundaries. In this large organization a search application was created to meet these goals. This paper focuses on the main design concepts of the second release of the search application, and how these were affected by experiences gained throughout the project. This design focused on simplicity and discoverability. Preliminary results show that the design is usable and that users find it easier to find the information they are looking for. A general increase in user satisfaction is also established.

## Categories and Subject Descriptors

H5.2 [Information interfaces and presentation]: User Interfaces

## General Terms

Measurement, Design, Experimentation,

## Keywords

Enterprise Search, Faceted Search, usage testing

## 1. INTRODUCTION

Every business day, employees need to access information stored in various enterprise applications and databases. Employees want one entrance to all corporate information. They often perceive the company intranet as one fuzzy cloud of information, while in reality it is a set of highly isolated information silos. Enterprise search is meant to address this need by providing access to relevant information and by consolidating ranking and presenting it properly. But how does one achieve this? The larger the organization the more divergent the information access needs.

Users within this large global organization have very different needs when it comes to finding information. Marketing employees complain that too much technical nonsense is embedded in the search results, while technical users say they are lacking technical depth in the available material. They all want access to all existing information in an environment where security is highly prioritized and information access strongly restricted. Quotes like these are common:

> *"We want to search in all information."*
>
> *"Google can search the whole internet so why can't we search our own intranet?"*
>
> *"Why can't our intranet be like Google?"*

But one user also said this:

> *"It is a lot harder to find that one exact thing that you are looking for than finding loads of general information on a subject on Google."*

And she was right. If a user cannot express what she is looking for in a good way it is much easier to find a lot of general information about the subject than that one document she read once (but forget what it was and who wrote it). Users need help defining what they are looking for. So how does one go about creating a useful enterprise search application?

## 2. COMMUNICATION

The first step the project team took was to assemble a group of people that were interested in enterprise search. These people included:

- A steering committee, and a group of stakeholders that provided the project team with valuable input.
- Pilot testers who took part in workshops and interviews before the launch of the first pilot and then evaluated the new pilot releases of the search application. These people also often acted as search ambassadors; spreading the word about the project to their colleagues.
- Beta testers, who took part in a beta test prior to the first launch of the application. The word about the new search application spread through the other three groups without much effort from the project team. Thanks to some communication material and the involvement of these groups approximately 1500 employees took part in the beta test prior to the launch.

A large survey was also conducted within the company with the purpose of collecting information about the state of Enterprise Search. After the first release of the search application a follow up survey was conducted in order to measure the results of the newly released application. This information was combined with data from the search logs to analyze the search behaviors and information needs of the workers within the organization and resulted in a list of prioritized areas that needed further improvement. Usability was one of those areas. So how would you go about designing a usable enterprise search application for a global organization?

## 3. DESIGNING FOR ENTERPRISE SEARCH

Designing for enterprise search is challenging work that involves packaging complex functionality in an easy-to-use interface. Enterprise search is a means for companies to capitalize on their organizational knowledge. This can be done by helping users:

- Speed up their everyday tasks
- Discover new information
- Finding accurate information e.g. information they know they can rely on.
- Re-finding information they know exist, but cannot find
- Improving the opportunity for collaboration and sharing knowledge within the organization
- Find information that is relevant for them in their context or work.

How does this correlate to users desires for a Googlified intranet? It is not a coincidence that the verb "to google" has been added to several renowned dictionaries, such as those from Oxford and Merriam-Webster. Search has been the de facto gateway to the Web for some years now. And the users say: *"Give us something like Google or better."* This is what we chose to call the Google effect on user expectations.

Due to the highly complex information needs in the organization the project team initially had a vision to create the "one application to rule them all", the Enterprise Search Application. The application should be easily converted into a desktop application all users could have on their desktop. The search application would be the starting point for everything on the intranet and all users should go to the search application for information. But of course users already had their accustomed behavioral patterns and ways of finding information.

The first release of the new search application was a great improvement to what had been available earlier. Employees claimed that they were able to find the information they were looking for! But something still did not work. Many users found the application overwhelming and complicated to use. All the embedded functionality made it slow to load. Questions about being more like Google still came up.

## 3.1 Usage Tests

The design team decided to do a larger series of usage tests on the newly released application. And as suspected, users did not use the search application as the one starting point for all information. Instead users were showing behavior that followed that of the berry-picking model [1]. Search was only one way of finding information. And users did not separate different search applications from each other. The central search application was seen as the same as the old database search in the document management system; they were all search.

The importance of having a simple and graphically appealing interface is well established within HCI literature [2]. Instead of the one central powerful application, the design team decided to change the design into a simple application that at first glance resembles Google. Combine this application with services that make it possible for other systems to use the search engine for searching within their own data, users can get quick access to "search" from where ever they are.

So how does one create an easy Google like application that still will meet all the complicated information needs of various user groups? The design team's answer was to make the advanced functionality simple by hiding the complexity in plain sight. Thus creating an application that looked really simple, but with fast and easy access to more complex functionality.

## 3.2 Speeding up Every Day Tasks

Query logs from the search application show that names of applications and products are over-represented in searches suggesting that these are important everyday tasks for the users of the system. These queries are considered as navigational searches rather than informational [3]. Using the Pareto principle [6] a query suggestion list was compiled from items represented in the search logs. When a user first enters the application and starts typing a query she gets a list of suggestions for matching items. Clicking on one of the items directly takes the user to the application or product she was looking for. Since no search against the index is done this functionality not only speeds up everyday navigational searches but also saves performance. A user can search for travel expenses not knowing the name of the travel expense application and she will get a suggestion for the system and navigate to it by the click of a button.

The information compiled from the query logs was also used for the purpose of creating quick links for further aiding people in finding commonly requested information. The quick links are also used to make sure that users found accurate information. So if a user searches for a name of a product the official approved product page will show up on top above blog posts or research documents for that product.

## 3.3 Refinding Information

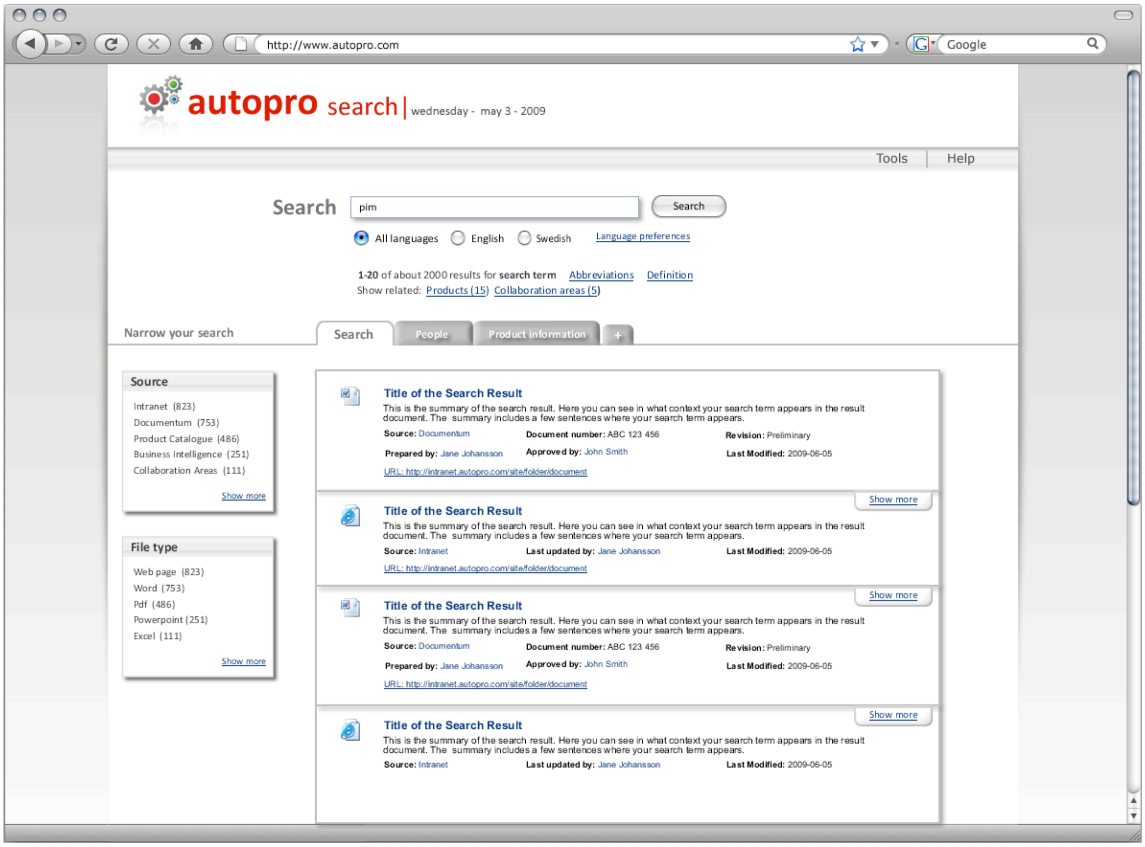
Research shows that people are very likely to revisit information they have viewed in the past and to repeat queries that they have used before [4]. Refinding information is important because many users know that the information exists on the intranet; they just do not know where it is located. Users even have problems finding their own documents. Functionality helping users to refind their information include:

- Searching my items, a quick way for users to find documents they are working on.
- Colleagues' items, a quick way of searching for documents written by a user's colleagues.
- Personalized search views where a user can choose to search within her part of the intranet where she knows the information is applicable to her situation. This will also help users feel that the information they find is accurate and appropriate.
- Bookmarkable URLs so users can save searches as bookmarks in the browser. A single search result can also be bookmarked from within the search application. Users can search within their bookmarked search results and the results are also marked with an icon in search results.

## 3.4 Personalization with Search Views

Users need to find information that is relevant in their context of work. An example of this is the needs to search for products and the relationships between versions of products as well as the related documentation and support material. There were many different examples like these where a particular scenario applied for a specific user group. So the design needed to incorporate both the general and the specific.

The design incorporated a general view of information where everything was searchable. This would be ideal for looking up general administrative information such as information about parental leave or holidays, or finding the official public information about the products sold by the company. The general search view is also a good way of getting an overview of all the information available on a topic. It resembles a standard Google style results list but incorporating a few extra features such as metadata tailored to the type of search result, icons for different types of search results and sources as well as a few standard facets for quick filtering of search results.



1-20 of about 2000 results for **search term** Abbreviations Definition
Show related: Products (15) Collaboration areas (5)

Source
Intranet (823)
Documentum (753)
Product Catalogue (486)
Business Intelligence (251)
Collaboration Areas (111)

File type
Web page (823)
Word (753)
Pdf (486)
Powerpoint (251)
Excel (111)

Title of the Search Result
This is the summary of the search result. Here you can see in what context your search term appears in the result document. The summary includes a few sentences where your search term appears.
**Source:** Documentum **Document number:** ABC 123 456 **Revision:** Preliminary
**Prepared by:** Jane Johansson **Approved by:** John Smith **Last Modified:** 2009-06-05
URL: http://intranet.autopro.com/site/folder/document

Title of the Search Result
This is the summary of the search result. Here you can see in what context your search term appears in the result document. The summary includes a few sentences where your search term appears.
**Source:** Intranet **Last updated by:** Jane Johansson **Last Modified:** 2009-06-05
URL: http://intranet.autopro.com/site/folder/document

Title of the Search Result
This is the summary of the search result. Here you can see in what context your search term appears in the result document. The summary includes a few sentences where your search term appears.
**Source:** Documentum **Document number:** ABC 123 456 **Revision:** Preliminary
**Prepared by:** Jane Johansson **Approved by:** John Smith **Last Modified:** 2009-06-05
URL: http://intranet.autopro.com/site/folder/document

Title of the Search Result
This is the summary of the search result. Here you can see in what context your search term appears in the result document. The summary includes a few sentences where your search term appears.
**Source:** Intranet **Last updated by:** Jane Johansson **Last Modified:** 2009-06-05

**Figure 1. Overview of the search application. Note that the image is an example and does not portray the real application**

The design also incorporates several different search views targeted for a specific user group, represented in the GUI as tabs. Users on a local branch in South America could search only within their part of the intranet, customer support could search within all support documentation as well as lessons learned from other projects from all over the world. A set of predefined views on the internal information was created for this purpose. Early adopters or department managers could also set up specific views on the information and in just a few seconds share them with their coworkers.

This provided search in all information for all but also fitted the personal needs of a large number of user groups. Users could find the information that they needed in their context of work. They could also easily share these views with their colleagues.

## 3.5 Presenting Search Results

The search results list is the essence of the search application. Presenting proper information about the results is essential for meeting the goals set for the application. The new results list in many ways looked like Google. But the design team wanted to find a way for users to discover the possibilities available with the new powerful search tool. The search results list was therefore fitted with an expanding function, where users with the click of a button could see more information about a search result. The expanded result included:

- More metadata about results where some of the metadata were links. The user could then directly access the system, site or information about the author or a product. This increases discovery of new related information and also speeds up everyday tasks.
- Icon displaying bookmarked results for easily refinding information.
- Advanced ways to filter the search on the specific type of result, or search within a specific site or subset of information directly from the search result.
- Links to related information and functionality within the different source systems such as approved versions of a document.
- Preview of the contents of documents so users can have a look at the document and read content directly, without having to enter another application or download and open a specific program such as Microsoft Word or Adobe reader for pdfs.

- Information about related products, collaboration areas, abbreviations and definitions were also displayed above the result list to further aid the users quest for information. The related information helps users discover new information they did not know existed.

## 3.6 Improving Opportunity for Collaboration

Related information also included collaboration areas that match a users query thus helping users discover new communities or opportunities for collaboration that they might not have known existed. Collaboration was also aided simply by making the collaboration areas searchable in the search application. Search provided easy access to the collaboration areas, even for those who have not started using them yet.

Users can easily share searches through bookmarkable URL:s. They can also share their search views and customizations with their colleagues.

The result was an application that seemed very simple at first glance, but still included all the different functionality needed in order to fulfill the information needs of the organization's different user groups. The new design was evaluated through usage test and though it included the same functionality as the old search application the results were completely different. Users found it not only easier to use but also easy to discover new information. They found it easier to determine whether a search result was interesting or matched what they were looking for, which minimizes the behavior of pogo-sticking [5]. The facets options that have not been understood or used previously were highly appreciated. All in all this confirms the importance of a simple and graphically appealing design.

# 4. CONCLUSIONS

The search application described in this paper had an overall positive impact on findability for this company. 9 out of 10 users use it every week. But even though the search tool has improved a great deal, there is still room for further enhancement in order for the company to fully capitalize on the investment in Enterprise Search. The improvement areas include:

- More functionality tailored to a specific scenario or user group in detail.
- Embedding the search functionality in other IT systems.
- Include even more information sources to assure that the search application includes all necessary information sources.
- Focus on contextualizing and facilitating local search. The search application needs to take into consideration the users geographic as well as organizational location, and also their role/business process in the organization in order to filter and rank results according to the users context.
- Continue to focus on the usability and performance of the search application.
- Further work on the communication about the new application is also needed to inform even more employees about the value of the new search application.

# 5. ACKNOWLEDGMENTS

Our thanks to all the people who took interest in our project and participated in interviews and usage testing to help improve the enterprise search application.

# 6. REFERENCES

[1] Bates, Marcia J. 1989 The Design of Browsing and Berry-picking Techniques for the Online Search Interface Online Review 13 (October 1989): 407-424.

[2] Hearst Marti, A. 2009 Search User Interfaces. Cambridge University Press ISBN 9780521113793

[3] Jansen, Bernard J. 2008 Determining the Informational, navigational and transactional intent of web queries. Information Processing and Management 44 (2008) 1251–1266

[4] W. Jones, S. Dumais, and H. Bruce. Once Found, What Then? A Study Of Keeping Behaviors In The Personal Use Of Web Information. *Proceedings Of The American Society For Information Science And Technology*, 39(1):391–402, 2002.

[5] Spool Jared M. 1999 Web Site Usability: A designer Guide

[6] http://en.wikipedia.org/wiki/Pareto_principle

# Cultural Differences in Information Behavior

Anita Komlodi
Department of Information Systems,
UMBC
1000 Hilltop Circle
Baltimore, MD 21250
USA
1 410 455 3212
komlodi@umbc.edu

Karoly Hercegfi
Department of Ergonomics and
Psychology, Budapest University of
Technology and Economics
Egry Jozsef utca 1.
H-1111 Budapest, Hungary
36 1 463 2654
hercegfi@erg.bme.hu

## ABSTRACT

With the availability of online translation services and the large amount of English-language content on the Web, more and more global users come in contact with content that was not created in their own language or culture. While some sites make efforts to localize their user interfaces and content, many simply translate content and use the same user interface. This is in direct contrast with findings that different cultures approach knowledge, information, and interaction with information in different ways. This paper will describe work in progress to study some national cultural differences in information behavior and the problems users face while interacting with information that was created in a language and culture different from their own.

## Keywords

Cross-cultural comparison, information behavior, user study

## 1. INTRODUCTION

The World Wide Web has integrated into the everyday information behaviors of many users. Information seekers often turn to the Web for solutions related to problems ranging from everyday life to health issues and professional information problems. The availability of large amounts of content from developed countries on the Internet and faster and faster network speeds create a global user group for many websites. While access to native-language content is constantly improving, it is often the case that Web users interact with information that was not originally created in their language or cultures.

Many websites and search services often simply provide translated content without localizing the user interfaces or the format of the information. As an example, Figures 1 & 2 represent the Google interface in the US and in Spain. While the language is different, the layout, organization, and the content of the screens remain very similar.

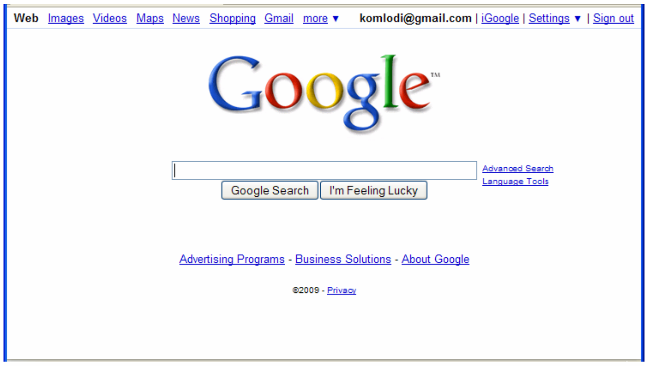


**Figure 1. The Google interface in the US.**

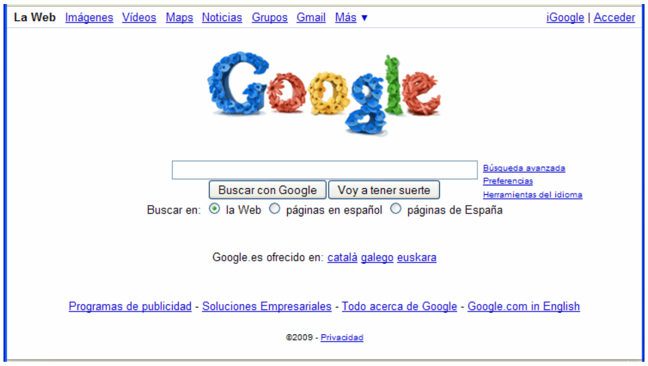

La Web
Imágenes
Vídeos
Maps
Noticias
Grupos
Gmail
Más ▾
iGoogle | Acceder
Google
Buscar con Google
Voy a tener suerte
Búsqueda avanzada
Preferencias
Herramientas del idioma
Buscar en: ● la Web ○ páginas en español ○ páginas de España
Google.es ofrecido en: català galego euskara
Programas de publicidad - Soluciones Empresariales - Todo acerca de Google - Google.com in English
©2009 - Privacidad

**Figure 2. The Google interface in Spain.**

Edward Hall, a leading cultural anthropologist of the 20th century wrote that "One of the functions of culture is to provide a highly selective screen between man and the outside world. In its many forms, culture therefore designates what we pay attention to and what we ignore." (Hall, 1976, p. 85) Culture can influence how we process and organize information and various cultural characteristics can influence many aspects of the information-seeking process (Komlodi & Carlin, 2004). This influence exists at the group and organizational levels as well; however, the focus of this paper is national culture.

One specific example is the case of search histories and different time concepts. The importance of search histories for search systems and user interfaces is often acknowledged (Hearst, 2009).

*Conference'04*, Month 1–2, 2004, City, State, Country.

In Western cultures most search history displays reflect event in a linear, temporally ordered list (Twidale, 1998; Komlodi, 2007). These displays reflect the Western culture's time concept where time is a linear, exhaustible resource (Hall, 1990). Hall (1990) defined monochromic and polychromic cultures. Monochromic cultures usually handle single tasks at a time organized in a linear fashion. Northern Europe, Northern America, Canada, and Australia fall into this group. In many other areas of the world, such as Latin America, Southern Europe, and Asia, time is a circular, renewable resource and people usually juggle multiple tasks at the same time (Hall, 1990). Do linear representations of task histories serve the needs of the users from polychronic cultures? Since most of this research, including user studies, design, and evaluation, originates in Western cultures, we do not have a good answer to this question.

Another example of cultural differences that impact humans' interaction with information is that of categorization. Categorization reflects how we understand and make sense of the world and it is strongly impacted by the cultural and educational system we grow up in. Cultural differences in classification systems and categorization have long been described (Clemmensen et al., 2009). The impact of various categorization systems on website design is very strong, as the foundation of the design of any website is the underlying categorization of the information, the information architecture. Users who are not familiar with the traditions of information organization in the culture of the website can face difficulties while navigating the site.

Thus, the design of sites that provide information access to global user groups should consider these cultural differences. However, to understand the need for localization, we first need to understand the strongest culturally influenced differences in Web information seeking behavior.

## 2. RELATED RESEARCH

Comparisons of information-seeking behavior across cultures are limited. Studies in information science look at international user groups in Western cultures, such as international students (Park, 1997, Yi, 2007) or immigrants (Fisher et al., 2004). These studies take cultural differences into account, however, they examine groups that are in a sense bicultural or in the process of becoming bicultural by integrating into a new culture. They also often examine the specific needs of the group but do not necessarily compare them to another cultural group. These studies are often conducted as needs assessment efforts for library or other public information services and examine behavior in the context of these institutions.

Very few studies looked at cross-cultural comparisons in information seeking outside of these needs studies in information science. However, the studies that took place found interesting differences. Iivonnen and White (2001) showed varying levels of cultural difference in information seeking on the Web among Finnish and American students. They focused on the choice for initial search strategies, not the full search process. Duncker (2002) examined the differences in searching via the online public access catalog of a library between indigenous and European New Zealanders and found significant differences in their interpretations and perceptions of the library. Duncker's study provides an interesting insight into New Zealand's Maori population's attitudes toward and relationship to information, more specifically information encoded and preserved in a library. Their notion of information sharing and capture are dramatically different from the library's view of collection, organization, and preservation, which creates conflicts between the library and its Maori patrons. This study is an important example of the complexities of the impact of culture on a user's attitudes toward and interactions with information.

Both of these studies attributed differences to various cultural differences: searching style, cognitive style, language use, perceptions of search systems (IIivonnen and White, 2001); and traditions of story telling and information sharing (Duncker, 2002). We will consider these intervening factors in our own study. Evers (2001) found differences in various culture groups' search, navigation, and user interface understanding within one website while studying differences in the use of a virtual learning environment. She attributed these differences to a set of variations in cultural characteristics. Yan, Finn, and Lu's (2007) studied international students at Virginia Tech but went a step further than other needs studies and specifically compared the American and international students' information behavior at Virginia Tech. They found differences in several areas such as preferences for initial information channels, library use frequency, familiarity with library services, and use of instructional resources.

Information systems researchers also often examine cultural differences surrounding information technology in business settings. Studies look at the impact of both national and organizational culture. We will focus on those that study national culture. A 2003 review (Ford, et al., 2003) found that most studies examined information systems (IS) management, while IS development, operations, and usage were often ignored as topics of research. This review focused on studies that incorporated Hofstede's (2001) ncultural dimensions. The category of IT usage studies is especially important for our study, as these describe user behavior related to various IT use areas, including information behavior. Leidner and Kayworth in 2006 found a different state of affairs when they extended their review to include all studies looking at cultural differences that presented new findings, and not just those using Hofstede's (2001) dimensions. They identified 30 studies that examined either IT use or outcome, the category where cross-cultural studies of information behavior may occur. Eighteen of these studies involved national cultural variables. An overwhelming result of these studies is that national culture makes an important difference in IT use and outcomes. As Web-based information seeking is carried out in the context of a specific technology, these results forecast important differences for information-seeking tasks as well.

Some of the studies more specifically involved tasks related to information seeking or other human-information interaction tasks, such as the larger domain of knowledge management. Chau et al. (2002) found that US users use the Internet mostly for information seeking, while their Hong Kong counterparts used it mostly for social communication and tied these finding to values related to the community versus individual achievement. Calhoun et al. (2002) found differences in how users in high and low context cultures process information. Hall (1990) defined high context cultures as those where much of the information content of a message in embedded in the context and not in the message itself.

Low context cultures express more of the information content in the message itself and do not rely on the context for the interpretation of the message. In Calhoun et al.'s 2002 study, users from a high context culture (Korea) were more easily overwhelmed by the information provided by their IS than users from the low context culture of the US. These differences further imply that a systematic cross cultural study of Web information seeking behavior will identify important differences.

## 3. PROPOSED RESEARCH

In this study we will address two main research questions related to the problem described in the introduction: 1) Do users in two different cultures exhibit different information behaviors on the Web and do they report different information-seeking habits? If yes, which phases and types of information behaviors are the most strongly impacted by culture?, 2) How does information seeking in the users' own language and for content created in their own culture differ from looking for information in a different language and created in a different culture? These broad research questions will drive an exploratory study of behavior to narrow down the impact of various user factors on search behavior.

Participants will be recruited in the US and Hungary. Participants with a significant knowledge of a second language will be recruited in order to allow for data collection answering the second research question. At the beginning of the session participants will be asked to fill out several questionnaires to collect data on the following independent variables: 1) a demographic questionnaire, including age, gender, cultural background, computer and Web experience, and Web information-seeking and use experience, 2) a cognitive style questionnaire, for example the Myers Briggs Type Indicator, 3) a cultural dimensions questionnaire, for example Hofstede's (2001) questionnaire to establish the participant's placement on various cultural dimensions.

The demographic questionnaire will help us confirm the homogeneity of our samples in terms of various demographic variables. The cognitive style questionnaire will inform our analysis by linking performance to cognitive style which has been shown to influence information seeking. In one of our previous studies (Hercegfi & Kiss, 2009), a specific aim of the series of experiments was to compare the behavior of users during solving information-seeking tasks. We were able to identify some significant differences between the behavior of users with different demographic backgrounds and cognitive styles. The new series of experiments will focus on the effects of a new dimension: the cross-cultural aspect. The application of the cultural dimension questionnaire will ensure that participants exhibit various characteristics typical of the groups they are members of.

Next, the participants will be asked to carry out several information seeking tasks. Some of the tasks will be prescribed for them, while others will be defined by the participants. The prescribed tasks will be representative of various information-seeking task types, including: known item seeking, subject driven and exhaustive searches. There will be no starting points defined for the tasks, as both Iivonen and White (2001) and Liao et al., (2007) found that there were strong cultural differences in terms of search starting points. At the end of the session the participants will be interviewed about their information-seeking experience and habits.

Objective parameters of the users' behavior during the session will be recorded, including their computer activity log, physiological data such as hear rate and skin conductance, and eye gaze movements. From the activity log we will record the following data: starting points, time spent on pages and steps, types of steps, number of steps, sequence of steps, and number of search results. We will use the physiological data to identify high mental effort steps (Hercegfi et al., 2009) and emotional reactions (Hercegfi et al., 2009). Both of these will supplement the analysis of the video capture of the participant's facial expressions and body postures. The eye gaze data will help us identify lower-level steps in the users' activities and hotspots on the pages that were particularly popular.

## 4. CONCLUSIONS

Previous studies have shown national cultural variations in information seeking and use behavior in various contexts. Most studies either looked at the needs of a specific user group, studied only a part of the information seeking and use process, or examined information behavior as a high level activity compared with other activities, but not the specific steps of the process. The proposed study will systematically study the information seeking process and identify those area most impacted by culture. While the results will be limited for those two cultures, other groups with similar cultural characteristics can also benefit from the findings. The results can also provide the basis for future studies involving more cultures. It is hoped that the results of the user study can provide guidance for the designers of information websites that serve a global audience.

REFERENCES

[1] Calhoun, K. J., Teng, J. T. C., Cheon, M. J. 2002. Impact of national culture on information technology usage behavior: An exploratory study of decision making in Korea and the USA. Behaviour and Information Technology. Vol. 21, No. 4. pp. 293-302.

[2] Chau, P. Y. K., Cole, M., Massey, A. P., Montoya-Weiss, M., O'Keefe, R. M. 2002 Cultural differences in the online behavior of consumers. Communications of the ACM. Vol. 45, No. 10. pp. 138-143.

[3] Clemmensen, T.; Hertzum, M.; Hornbaek, K.; Qingxin, S.; Yammiyavar, P. 2009. Cultural cognition in usability evaluation. Interacting with Computers. Vol. 21. pp. 212-220.

[4] Evers, V. 2001 Cultural aspects of user interface understanding. Doctoral Dissertation. Open University, London, England, p. 377.

[5] Duncker, E. 2002 Cross-Cultural Usability of the Library Metaphor, *JCDL '02,* Portland, Oregon 13-17 July 2002.

[6] Fisher, K.E., Durrance, J.C. & Hinton, M.B. (2004). Information grounds and the use of need-based services by immigrants in Queens , NY : a context-based, outcome evaluation approach. Journal of the American Society for Information Science & Technology, 55(8), 754-766.

[7] Ford, D. P.; Connelly, C. E.; Meister, D. B. 2003 Information Systems Research and Hofstede's Culture's

Consequences: An uneasy and incomplete partnership. IEEE Transactions on Engineering Management, Vol. 50, No. 1. pp. 8-25.

[8] Hall, E. T. 1976 Beyond culture, New York: Doubleday.

[9] Hall, E. T. 1990 The hidden dimension, New York: Doubleday.

[10] Hearst, M. 2009. Search User Interfaces. Cambridge University Press. Accessed online: http://searchuserinterfaces.com/

[11] Hercegfi, K., Csillik, O., Bodnár, É., Sass, J., Izsó, L. 2009 Designers of Different Cognitive Styles Editing E-Learning Materials Studied by Monitoring Physiological and Other Data Simultaneously. 8th International Conference on Engineering Psychology and Cognitive Ergonomics, HCII2009, San Diego, California, USA, 14-24 July 2009, Proceedings LNAI 5639, Springer, ISBN 978-3-642-02727-7, pp.179–186.

[12] Hercegfi, K., Kiss, O.E. (2009): Assessment of e-Learning Material with the INTERFACE System. In: Szűcs, A., Tait, A., Widal, M., Bernath, U. (eds): Distance and E-Learning in Transition. John Wiley & Sons and ISTE, Hoboken, NJ, USA. ISBN 978-1-84821-132-2, Chapter 45, pp.645-657.

[13] Hofstede, Geert H. 2001 Culture's consequences : comparing values, behaviors, institutions, and organizations across nations, Thousand Oaks, Calif. : Sage Publications.

[14] Iivonen, Mirja; White, Marilyn Domas. 2001 The choice of initial web search strategies: A comparison between Finnish and American searchers, *Journal of Documentation,* v57 n4, pp. 465-491.

[15] Komlodi, Anita. Carlin, Michael. (2004) Identifying cultural variables in information-seeking behavior. Proceedings of the Tenth Americas Conference on Information Systems (AMCIS). New York, NY, Association for Information Systems. Pp. 477-481.

[16] Komlodi, Anita; Marchionini, Gary; Soergel, Dagobert (2007) Search history support for finding information: User interface design recommendations from a user study. Information Processing and Management. Oxford, New York, NY, Pergamon Press. Vol. 43. Pp. 10-29.

[17] Leidner, D. E.; Kayworth, T. 2006 A review of culture in information systems research: Toward a theory of information technology culture conflict. MIS Quarterly, Vol. 30. No. 2. pp. 357-399.

[18] Park, I. 1997 A comparative study of major OPACs in selected academic libraries for developing countries – User study and subjective user evaluation. International Information and Library Review. Vol 29, pp. 67-83.

[19] Twidale, M.; Nichols, D. 1998 Designing Interfaces to Support Collaboration in Information Retrieval. Interacting with Computers, 10(2):177–193.

[20] Yi, Z. 2007 International student perceptions of information needs and use. The Journal of Academic Librarianship. Vol. 33, No. 6, pp. 666-673.

# Adapting an Information Visualization Tool for Mobile Information Retrieval

Sherry Koshman
School of Information Sciences
University of Pittsburgh
Pittsburgh, PA U.S.A.15260
+1 412-624-9441
skoshman@sis.pitt.edu

Jae-wook Ahn
School of Information Sciences
University of Pittsburgh
Pittsburgh, PA U.S.A.15260
+1 412-624-9437
jahn@mail.sis.pitt.edu

## ABSTRACT

The application of information visualization (infoviz) tools to mobile devices for information retrieval (IR) is uncommon. This has been attributed to the complex challenges related to mobile devices including the technical restrictions upon generating a small screen graphical visual representation for abstract information. This paper reports on a work in progress on the basic interface design and creation of MVIBE (Mobile VIBE), a new mobile version of VIBE (Visual Information Browsing Environment), which is an information visualization tool developed for information retrieval. MVIBE was developed and tested on the Apple, Linux, and Windows mobile platforms. User feedback was obtained and some of the reported challenges are common to mobile technology and others to general information visualization. At this early stage, the overall question is: can mobile devices be effective for generating a viable visualization of search results? The paper concludes with observations gained during the adaptation process, recommendations for the next phase of Mobile VIBE development, and future design considerations for developing information visualization interfaces on mobile devices.

## Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces – *prototyping, screen design.*

## General Terms

Design, Verification

## Keywords

Mobile Information Visualization, Mobile Information Retrieval

*HCIR09*, October 23, 2009, Washington, D.C., U.S.A..
Copyright 2009 ACM 1-58113-000-0/00/0004…$5.00.

## 1. INTRODUCTION

Mobile computing environments are engaging users for various information retrieval (IR) tasks due to the availability of wireless access, mobile browsers, improved mobile devices, and their ubiquity in the information landscape [2]. Factual queries dominate mobile IR for items such as weather, calendar information, and traffic reports. However, persistent access to the web is expanding and mobile search is becoming a prominent mode of interaction that users adapt from familiar desktop environments [10].

Text display on the small screen size of mobile devices is limited and information visualization tools offer potential in presenting concise representations of retrieved results to users. Mobile search has been the object of academic exploration; however the presentation of visualized mobile search results has not. Mobile IR is in its very early stage and information visualization can be integrated into the user's mental model of search result presentations since display options for mobile information retrieval are not yet firmly established.

The contribution of this paper is significant to mobile information retrieval visualization since it presents a technical case study of adapting the desktop VIBE interface to a mobile device. MVIBE (Mobile VIBE) is one of the first graphical information visualization tools for IR to be developed from its desktop version. In addition, this is one of the first studies to examine the new information visualization on three different mobile platforms using personal digital assistants (PDAs) devoid of telephony.

## 2. RELATED WORK

The predominant themes of previous work relating to this project include mobile information retrieval, mobile visualization interfaces, and mobile device testing for evaluating visualized representations.

**Mobile Information Retrieval**. Research encapsulates work on rendering web pages for mobile displays and the usability of web page layouts for various types of tasks [24,30,25,26}. Users' information seeking activities for web-based data are examined through mobile transaction log analyses and the approaches to mobile search offer avenues to apply visualization techniques to retrieved results [11,14,16,15,28].

Mobile users' browsing and searching patterns show that browsing is predominant, however query searching reflected lengthier interaction times and higher user interest [9,23]. Visual tag clouds are used for query representation in a system where users can view other user queries rather than search results for a particular geographic location in which they are situated [13]. Queries which were used most frequently for that location appear in a larger text size.

**Mobile Visualization Interfaces.** Initial work on visualized mobile web applications was directed toward resolving the end user's request for web-based location information. As a result, many mobile visualization studies focused on web or GPS (Global Positioning System) based mobile mapping data to aid users' navigation tasks [3,5]. Digital mobile maps represent geographic information and networked routes that are a direct analog to the physical environment.

Research in mobile information visualization for information retrieval is limited. The lack of robust research may be attributed to its focus upon visualizing non-physical abstract information and the restrictions of mobile technology [7]. However, visualizing abstract document spaces on mobile screens is discussed by [17,29]. To address the issue of high information density on small screen, [27] developed a graphical interface based on "liquid browsing", an animated scatter plot that resembles suspended bubbles in a liquid which is most suited to pressure sensitive screens that are not common among handheld devices. Work on visualized screen layouts, node occlusion, and underutilized screen space issues have mixed results [29,12,4,6].

**Mobile Devices.** With exception of the iPod touch, Nokia and Hewlett Packard (HP) devices are primarily used in previous mobile visualized research as prototypes [29,4,6]. Work conducted specifically on personal digital assistants was shown for search result clustering and the scatterplot prototype [4,6].

## 3. DESKTOP VIBE

VIBE (Visual Information Browsing Environment) is a desktop prototype system developed by researchers at Molde College, Norway and at the School of Information Sciences, University of Pittsburgh. A more current desktop version was built as a Java Swing (or applet). Its original implementation had a strong impact on the direction of information visualization in IR for over a decade [22,18,8,21,19,20,1]. The selection of VIBE for this work was based on its wide implementation that reduced the uncertainty associated with information visualization system use.

Basic elements of the VIBE interface include a visualized query that has round circular icons which represent Points of Interest (POIs) or user selected terms from a drop down menu of options. The resulting document set is depicted as polygons that are plotted in proximity to the POIs according to a term frequency distribution algorithm (Figure 1). The larger the document icon, the more frequent occurrence of terms related to the document. Desktop VIBE has a robust set of features to manipulate the visualization. For example, color is prominently used to mark POIs and to indicate overlapping retrieved documents. POIs may be dragged, added or removed from the display. Relationships between the POIs and the documents can be depicted using the "lines" or "net" features The salient premise in VIBE's design is that the query and resulting document set can be visualized in one screen for user browsing and item selection.

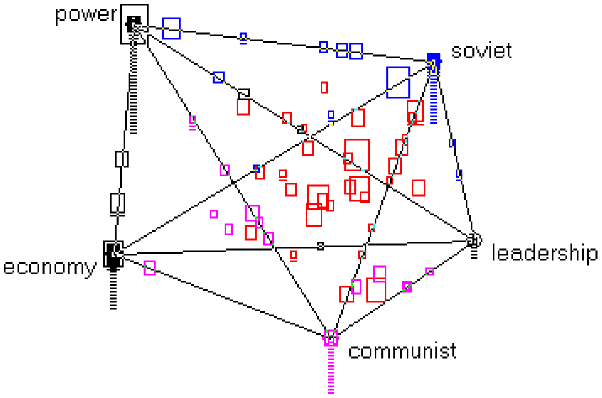


**Figure 1: Desktop VIBE Display.**

The first comprehensive usability study of the desktop VIBE system showed that its interface and availability of robust features for IR made it a prime candidate for information visualization adaptation in the mobile environment [19,20].

## 4. MOBILE VIBE (MVIBE)

The design of Mobile VIBE began with a review of the initial user study with desktop VIBE [19,20]. It was observed that while desktop VIBE had well developed interface features, they could not all be deployed simultaneously in a mobile visualization.

**Task 1: Defeaturing**. The first objective in creating Mobile VIBE was to defeature the desktop VIBE interface by selecting interface options to yield a simpler and clearer mobile interface for end users [21]. Original desktop system options such as "star", "lasso", "astro" or "Boolean" views were not included in the mobile design. Colors were limited to green for the POI display, red for their moveable state indicating that POIs are selected for the user to relocate them, blue for the "lines" feature and document icons, and a white background for the display.

Salient features were selected and applied to the mobile version on the basis that they were amenable to a mobile screen display and function. The list includes three system-generated default display items that enable data coding and seven user controlled features to use in the display (Table 1).

**Table 1: Mobile VIBE Interface Features**

| Mobile VIBE Default Features |
|---|
| **POIs.** Points of Interest, which are document terms represented as circular icons on the display. |
| **POI Labels.** Text labels that show the query terms above the POIs (Points of Interest). |
| **Document Icons.** Squares on the display that represent retrieved documents. |
| **Mobile VIBE User Display Options** |
| **Show Document Titles.** Allows the user to make visible the text derived from the search results positioned near the document icons. |
| **Begin Dragging.** Notifies the user of the initial POI movement. |
| **End Dragging.** Notifies the user of the POI movement's end. |
| **Show Lines.** Allows the user to display axes between two terms |

| |
|---|
| (POIs). All document icons situated on the line are related to the terms on either end. It is useful for determining the influence of POIs on document icons. |
| **Move POIs.** A move is activated by clicking on a POI and then clicking on another part of the screen to place it. |
| **Color.** Associated with the *Move POIs* feature. The default POI color is green and the color changes to red while the user relocates the POI on the screen. |
| **Reset POIs.** Places POIs in the default circular arrangement. |

**Task 2: Mobile Development.** Mobile VIBE was authored in JavaScript in order to make the code multi-platform compliant and to take advantage of a wider range of support from various mobile devices. Unlike the desktop environment, Java applets are not fully supported by mobile devices at this point, whereas JavaScript is more ubiquitous. Mobile VIBE is a JavaScript port of the Java Swing version of VIBE. Due to the similarity of Java and JavaScript syntax, the adaptation was relatively straightforward and was mostly done by direct translation. The porting became possible due to recent support of the CANVAS tag from state-of-the-art Web browsers, such as Safari, Firefox, and Opera. The CANVAS tag allows the manipulation of every pixel in the area it occupies, so it was possible to freely draw the VIBE visualization on it.

Data were encoded on Mobile VIBE in a consistent manner with its predecessor. Document icons are geometric squares, however the icon size to represent term frequency of the POIs was not implemented in the mobile version. Figure 2 shows an example of the graphical Mobile VIBE interface on the HP iPAQ running Opera 9.5 Beta on Windows Mobile 6. The circular green icons represent the terms or POIs, term labels appear above the icons and the lines feature is activated to connect the terms on the display. The square icons are the retrieved items and the interface options are presented at the bottom of the display. In this example, the visualization is highlighted with square borders. The portrait view is shown although the device may be used in a portrait or landscape position to view the visualization on the HP iPAQ and iPod touch.

Since all three of the mobile Web browsers used in the current investigation support JavaScript and the CANVAS tag, the drawing of the visualization in the mobile environment was done without any modification of the desktop version of the VIBE JavaScript code. However, due to the different input methods they support, the interaction mechanism needed to be updated. The code was loaded on to three representative personal digital assistants, which are described next.

**Task 3: Mobile Interaction**. The three PDAs, their screen resolutions and browsers include: the iPod touch running iPhone OS 2.1.1 (480x320) and Safari 3.1.1; the Nokia N810 running Maemo Linux 4.1 OS2008 (800x480) and the Mozilla-based MicroB 1.0.4 browser; and the HP iPAQ 211 Enterprise Edition (640x480) running Windows Mobile 6 and the Opera 9.5 Beta browser.

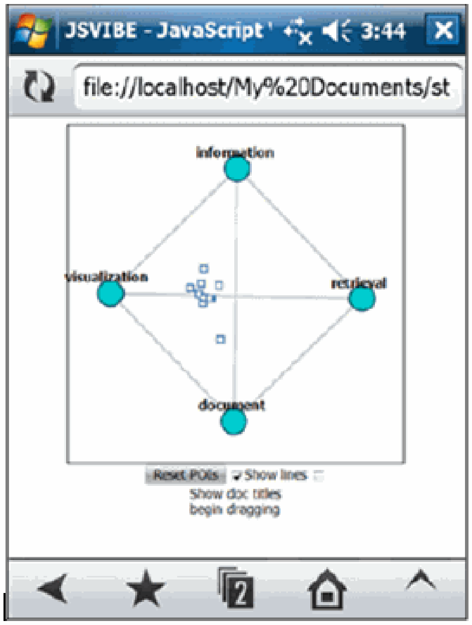


**Figure 2: MVIBE Interface on the HP iPAQ.**

Apple's iPod touch supports the most unique input method with its multi-touch interface, which allows users to use their two fingers to click, drag, squeeze, and rotate objects on the screen. Apple provides a simple set of JavaScript event handling functions for the multi-touch events and the mouse dragging action of the desktop version could be easily adapted to the new touch-based POI dragging. The Nokia and HP machines use stylus pens as input devices and they could drag the POIs without modification of the original VIBE JavaScript code. However, it was difficult to drag the POIs with the stylus pen due to the smaller size of the mobile screen and the viewport panning nature of the devices (the screen itself frequently panned around while dragging the POIs).

Therefore, a simpler method for moving POIs was added: a click-move-and click again function. With this functionality, when a user clicks on a POI, its color changes to red, showing that it is in a movable state. Then the user moves his/her stylus pen to any arbitrary position on the screen (not dragging) and then clicks on it. The POI, which was just selected, jumps to the new position.

Four query terms were represented as four POIs on the screen as green circles and the documents were displayed as small squares equipped with their titles (Figures 2 and 3). The most noticeable problem was the clutter of visualization elements such as the document titles and lines among the POIs (Figure 3a). To resolve the visual clutter, options to turn off the lines and titles were added (Figure 3b). Moreover, users can freely move the POIs and relieve the clutter of the document titles (Figure 3c).

A small sample data set was constructed and visualized. The data are a single web search result, generated by the Google web search engine. The query, "information visualization document retrieval", was entered on Google and the top 10 documents returned by Google were downloaded. The 10 web pages were indexed using the Indri search engine (www.lemurproject.org/indri) and the probability values between each document and each query term, "information", "visualization", "document", and "visualization" were calculated. The VIBE engine can then read the probability values and make use of the probability ratios for the visualization.

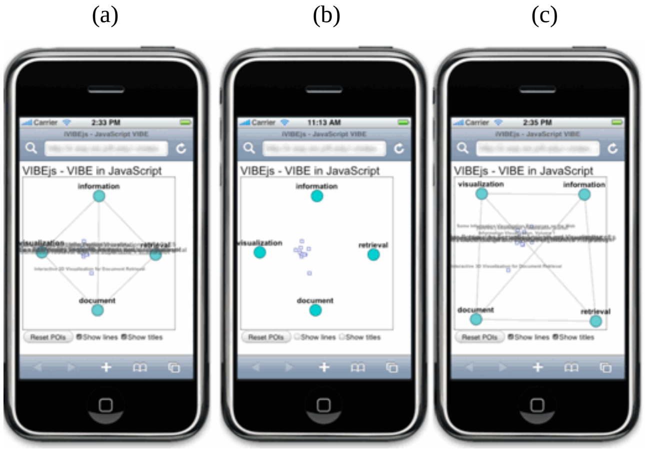


**Figure 3: MVIBE on the iPod touch.**

## 5. EARLY USER FEEDBACK

User feedback on the Mobile VIBE interface was elicited at this very early stage. Six doctoral students (four females and two males) from the School of Information Sciences volunteered to critique the Mobile VIBE interface. All but one had over one year of mobile device experience and had been exposed to information visualization. Their feedback is shown in Table 2.

**Table 2: User Feedback Summary**

| |
|---|
| **MVIBE Display Legibility.** Received positive user responses. |
| **Mobile Device Type**. Affected POI label identification and users ranked this task's effectiveness on the mobile devices in descending order: the iPod, Nokia, then the HP. |
| **Readable Document Titles**. Received low user ratings according to the same ranking of mobile devices as shown above. Occlusion is problematic. |
| **Interface Icon Visibility**. Positive user responses, except on the HP. |
| **Lines Feature**. Uniform favorable response by users, except on the HP. |
| **Move POIs**. Mixed success reported by users across devices. |
| **Resize Display**. Reportedly worked on the Nokia and HP, not the iPod. |
| **Understanding MVIBE features**. Positive responses received from most users. |

The users' observations of MVIBE on the iPod touch included the difficulty with screen control, how large finger size may be prohibitive to its use, how it was difficult to move the POIs, and how the multi-touch zoom does not work with Mobile VIBE. The users' favorite activities with MVIBE on the iPod touch included interacting with the display orientation, moving POIs, and using the "Reset POI" option. On the HP iPAQ, the preferred activity was the zoom functionality.

## 6. CONCLUSIONS AND FUTURE WORK

This technical case study presents an initial venture into the adaptation of an established desktop visualization tool designated for information retrieval, to a mobile device. The recommended steps to accomplish this goal include: 1) using a **defeaturing** technique to initially select few interface options for reducing visual complexity; 2) selecting a **cross-platform** development tool to address multiple mobile environments (e.g. JavaScript); 3) keeping visualization design principles aligned with mobile **interaction** techniques; and 4) obtaining **user feedback** at an early stage to facilitate incremental development

The initial step of defeaturing an infoviz interface is important for establishing a baseline for development. Testing designs on multiple platforms reveals how issues may appear in one platform, but not another. This is significant for users' selecting multiple device types. It was observed that users experienced different levels of difficulty when interacting with the visualization due to browser function interference (e.g. zooming). For this reason, browser-based mobile visualization interaction design needs to be reformulated to correspond with current mobile browser techniques. The visualization functionality may require adaptation to remain compatible such as changing the "move POI" MVIBE option from drag and drop to point and click (See Section 4: Task 3). Balancing mobile functions with the visualization interface design requires further work.

MVIBE is a work in progress and future work is planned. Users had positive responses in understanding MVIBE's interface features. Most graphical features were evident in the display however, the textual document titles' appearance and lines occlusion posed difficulty. Techniques, such as rollovers, are being considered to address the titles issue. What is surprising is the difference of interaction difficulties among the three devices. It is known that the variability of platforms among mobile devices is a larger functionality issue in comparison to the desktop environment, however how it specifically affects a mobile visualization is a new avenue for exploration. Future research will concentrate on fixing the movement, selection, zooming, resizing issues since incongruities were found among the devices and the visualization system. Additional IR features will be incrementally added and tested with a larger data set and more participants to evaluate the visualization in the mobile IR process. A personalization option will be investigated.

The early testing of Mobile VIBE shows promise for the field of mobile infoviz for information retrieval. The MVIBE case study demonstrates several factors to address when creating IR visualizations for mobile devices. It presents an affirmative answer to the initial guiding question in that mobile devices can be effective for generating a graphical IR visualization. Manufacturers' mobile products are improving and mobile challenges are becoming less onerous to overcome in order to generate future IR visualizations in the rapidly expanding mobile environment.

## 7. REFERENCES

[1] Ahn, J., Brusilovsky, P. and Sosnovsky, S., QuizVIBE: accessing educational objects with adaptive relevance-based visualization. In World Conference on E-Learning, E-Learn Proceedings, (Honolulu, Hawaii, 2006), Association for the Advancement of Computing in Education (AACE), 2707-2714.

[2] Ballard, B. Designing the Mobile User Experience. John Wiley & Sons, Chichester, England, 2007.

[3] Burigat, S., Chittaro, L. and Gabrielli, S., Visualization and multimodality: visualizing locations of off-screen objects on mobile devices: a comparative evaluation of three approaches. In Proceedings of the 8th Conference on Human-Computer Interaction with Mobile Devices and Services, (Helsinki, Finland, 2006), ACM, 239-246.

[4] Buring, T. and Reiterer, H., Input and visualization: ZuiScat: querying and visualizing information spaces on personal digital assistants. In Proceedings of the 7th International Conference on Human Computer Interaction with Mobile Devices & Services, (Salzburg, Austria, 2005), ACM, 129-136.

[5] Carmo, M.B., Afonso, A.P. and Matos, P.P., Query results and processing: visualization of geographic query results for small screen devices. In Proceedings of the 4th ACM workshop on Geographical information retrieval, (Lisbon, Portugal, 2007), ACM, 63-64.

[6] Carpineto, C., Mizzaro, S., Romano, G. and Snidero, M. Mobile information retrieval with search results clustering: prototypes and evaluations. Journal of the American Society for Information Science and Technology, 60 (5) 2009, 877-895.

[7] Chittaro, L. Visualizing information on mobile devices. Computer, 39 (3) 2006, 40-45.

[8] Christel, M. and Huang, C., SVG for navigating digital news video. In Ninth ACM International Conference on Multimedia, (Ottawa, Canada, 2001), ACM, 483 - 485.

[9] Church, K., Smyth, B., Cotter, P. and Bradley, K. Mobile information access: A study of emerging search behavior on the mobile Internet. ACM Transactions on the Web, 1 (1) 2007, 1-38.

[10] Costa, C.J., Silva, J. and Aparcio, M., Evaluating web usability using small display devices. In Proceedings of the 25th Annual ACM International Conference on Design of Communication, (El Paso, Texas, USA, 2007), ACM, 263-268.

[11] Cui, Y. and Roto, V., How people use the web on mobile devices. In International World Wide Web Conference, (Beijing, China, 2008), ACM, 905-914.

[12] Ghinea, G., Heigum, J. and Fongen, A., Information visualization for mobile devices: a novel approach based on the MagicEyeView. In Wireless Pervasive Computing, 2008. ISWPC 2008. 3rd International Symposium, (Santorini, 2008), 566-570.

[13] Jones, M., Buchanan, G., Harper, R. and Xech, P.-L., Questions not answers: a novel mobile search technique. In Proceedings of the SIGCHI Conference on Human factors in Computing Systems, (San Jose, California, USA, 2007), ACM, 155-158.

[14] Kamvar, M. and Baluja, S., A large scale study of wireless search behavior: Google mobile search. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, (Montreal, Quebec 2006), ACM, 701-709.

[15] Kamvar, M., Kellar, M., Patel, R. and Xu, Y., Computers and iPhones and Mobile Phones, oh My! In International World Wide Web Conference, (Madrid, Spain, 2009), ACM, 801-810.

[16] Karlson, A., Robertson, G., Robbins, D., Czerwinski, M. and Smith, G., FaThumb: a facet-based interface for mobile search. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, (Montreal, Quebec, Canada, 2006), ACM, 711-720.

[17] Karstens, B., Kreuseler, M. and Shumann, H., Visualization of complex structures on mobile handhelds. In Proceedings of the International Workshop on Mobile Computing, (2003).

[18] Korfhage, R.R. Information Storage and Retrieval. John Wiley & Sons, New York, 1997.

[19] Koshman, S. Testing user interaction with a prototype visualization-based information retrieval system. Journal of the American Society for Information Science and Technology 56 (8) 2005, 824-833.

[20] Koshman, S. A usability study comparing a prototype visualization-based system with a text-based system for information retrieval. Journal of Documentation 60 (5) 2004, 565-580.

[21] Morse, E., Lewis, M. and Olsen, K. Testing visual information retrieval methodologies case study: comparative analysis of textual, icon, graphical and "spring" displays. Journal of the American Society for Information Science and Technology 53 (1) 2002, 28-40.

[22] Olsen, K.A., Korfhage, R.R., Sochats, K.M., Spring, M.B. and Williams, J.G. Visualization of a document collection: the VIBE system. Information Processing & Management 29 (1) 1993, 69-81.

[23] Roto, V. Search on mobile phones. Journal of the American Society for Information Science and Technology, 57 (6) 2006, 834-837.

[24] Roto, V., Popescu, A., Koivisto, A. and Vartiainen, E., Minimap: a web page visualization method for mobile phones. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, (Montreal, Quebec, Canada, 2006), ACM, 35-44.

[25] Simon, R. and Frolich, P., A mobile application framework for the geospatial web. In International World Wide Web Conference (IW3C2), (Banff, Alberta, 2007), 381-390.

[26] Spence, R. Information Visualization: Design for Interaction, Second Edition. ACM Press, New York, 2007.

[27] Waldeck, C. and Balfanz, D., Mobile liquid 2D scatter space (ML2DSS). In Eighth International Conference on Information Visualization (London, UK, 2004), 494-498.

[28] Yi, J., Maghoul, F. and Pedersen, J., Deciphering mobile search patterns: a study of Yahoo! mobile search queries. In International World Wide Web Conference, (Beijing, China, 2008), ACM, 257-266.

[29] Yoo, H.Y. and Cheon, S.H., Visualization by information type on mobile device. In Proceedings of the Asia Pacific Symposium on Information Visualisation - Volume 60, (Tokyo, Japan, 2006), Australian Computer Society, Inc., 143-146.

[30] Zhang, D. Web content adaptation for mobile handheld devices. Communications of the ACM, 50 (2) 2007, 75-79.

# A Theoretical Framework for Subjective Relevance

Katrina Muller
SILS
University of North Carolina
Chapel Hill, NC 27599-3360 USA
+1 919 259 6832
kmuller@email.unc.edu

Diane Kelly
SILS
University of North Carolina
Chapel Hill, NC 27599-3360 USA
+1 919 962 8065
dianek@email.unc.edu

## ABSTRACT

This paper explicitly models subjective relevance by deconstructing its elements. We outline the various dimensions of subjective relevance, considering internal and external factors as well as interactions. We employ a utility framework for modeling, both conceptually and mathematically, subjective relevance and its multiple dimensions, aspects and interactions.

Categories and Subject Descriptors

H.1.2 [**Information Storage and Retrieval**]: Models and Principles – User/Machine Systems – Human factors

## General Terms

Human Factors, Theory

## Keywords

affective, cognitive, judgment, relevance, user-centered, utility.

## 1. INTRODUCTION

Although it is well documented that relevance is multidimensional and dynamic, most studies still assess it with a single measure at a single point in time [3]. Moreover, relevance is assessed in the same way regardless of the user's information seeking task, interest or personal characteristics. This leads to inquiries about what type of relevance is being modeled in information seeking studies, what is being measured and what criteria users employ when deciding which information objects are relevant to them. While simplifications and abstractions are often necessary in order to study phenomenon, this does not mean that efforts should not be made to better understand and measure relevance. As Saracevic notes, "relevance is a, if not *the*, key notion in information science in general and information retrieval in particular" (p. 1915) [15]. Given its centrality and importance (indeed, some notion of relevance is used in almost all human computer information retrieval studies), increased effort should be made to model and measure relevance.

In previous work, we discussed some of the limitations of current conceptualization and operationalizations of relevance in information retrieval [10]. In this paper, we present a conceptual model for subjective relevance and demonstrate a strategy for operationalization. This paper proposes utility as both a measure of, and as a theoretical framework for, subjective relevance. This approach has been presented by other researchers [4,12]. We believe the concept of utility is useful because it considers relevance from the user's point of view. In this paper we consider three types of user-centered relevance: cognitive relevance, affective relevance, and situational relevance [14,15]. We then present a conceptual and mathematical model of these three types of relevance and discuss interactions among them.

## 2. RELEVANCE AND UTILITY

Researchers have used many terms to describe subjective relevance or different aspects of it [1,4,8,14,16,18]. The number of these possible aspects is innumerable, as are the potential interactive effects of those aspects. Although this can present serious problems for measurement it also gives researchers leeway to construct measures appropriate for testing their hypotheses about relevance judgments and information seeking behavior.

Considering its historical development, utility seems like a good general concept on which to ground any discussion of subjective relevance. In Jeremy Bentham's *An Introduction to the Principles of Morals and Legislation* [2] he postulated that people should and do make decisions in order to maximize their pleasure while minimizing their pain. Bentham discusses the idea of happiness which is a pleasurable state of mind that comes from success or attainment of what is good [7]. By this definition, Bentham's greatest happiness principle is a necessary and sufficient condition for a criterion of relevance judgments. Bentham and his contemporaries used utility as a measure of this happiness or satisfaction.

Some of the terms used in reference to relevance are pertinence, psychological relevance, situation relevance, usefulness and value. All of these, and any others that come to mind, can be included in the definition of happiness above. Therefore maximizing utility (a measure of happiness) is synonymous with maximizing any manifestation of subjective relevance.

Cooper was one of the first researchers in IR to consider utility as a measure of retrieval effectiveness. His seminal work lays the foundation for our current discussion [4,5]. By challenging traditional, system centered views of relevance Cooper opens the door for researchers to provide rich theories of relevance that not only reflect algorithmic and topical forms of relevance but those that consider the user's needs and situation as well [1,8,14,16,18]. Our work synthesizes these theoretical constructs of subjective relevance into a cohesive framework using a utility model. Our goal here is only to present the theoretical framework, a validation of appropriate instruments and experiments testing the theory are planned for future work.

Having placed subjective relevance in a utility framework we next examine the choice variables that determine users' preferences. It is the direct and indirect effect of these aspects, their interactions with each other and external factors that describe the strategies and motivations which affect users' preferences and the subsequent relevance outcomes.

## 3. DIMENSIONS OF RELEVANCE

Saracevic conceptualized relevance along five dimensions: (1) system or algorithm; (2) topical; (3) pertinence or cognitive; (4)

situational; and (5) motivational or affective [12, 13]. *System* or *algorithm relevance* describes the relationship between a query and the collection of information objects. This type of relevance is operationalized by a particular algorithm, and does not involve user judgment. *Topical relevance* is associated with the aboutness of a particular document. For instance, if the user's query is 'elephants,' then a document containing a discussion of elephants is topically relevant. *Pertinence*, or *cognitive relevance*, describes the relationship between a user's perception of his information need, what he currently knows about the information need and a document. This is very much related to psychological relevance [6], which considers the degree of cognitive transformation or learning that is caused by reading a document. *Situational relevance*, originally coined by Wilson [18], is concerned with the idea that relevance judgments change according to task and situation. Finally, *motivational* or *affective relevance* considers the intentions, goals, motivations and emotions of the user.

Cognitive relevance and affective relevance are internal processes. Situational aspects usually start out as external conditions which eventually have an impact on internal processes. Classifying these relevance dimensions into internal and external categories is an important step toward the development of a viable theory of subjective relevance. Equally as important is considering the interaction between and among these internal and external factors. The effects that situational factors have on cognitive and affective relevance as well as the effects that these inward processes have on each other impacts how people make relevance judgments. To be clear we distinguish between internal processes and external conditions below.

### 3.1 Internal Processes

The internal aspects of subjective relevance—cognitive and affective—are constructs which do not exist outside the user. Cognitive relevance is the perception that the user develops of the information object given his or her cognitive state [14,15]. It is critical judgment and rational decision making. It is the recognition of an information need and a strategy to meet that need. Affective relevance, on the other hand, is the user's inner emotional state. It includes hopes, dreams and desires as well as goals and motivations.

### 3.2 External Conditions

Situational aspects of relevance originate outside of the user but can have serious impact on his or her internal state. To illustrate this, we will use one well-studied aspect of situation that has been found to impact relevance behavior, information task. We will use this example throughout out the remainder of this paper. Many researchers have studied how task affects relevance outcomes [3,11,19]. Some tasks can be quite cognitively demanding, while some are affectively charged. Different users have greater motivations for working on some tasks than others. Internal emotional and cognitive states are influenced by various situational factors including the environment and a user's time constraints.

As previously mentioned the relationships between and among all dimensions of subjective relevance are extremely important to the understanding of the formation of relevance judgments. We posit that affective and cognitive relevance are strongly interdependent, that task effects are manifest through either or both of these aspects and that preferences over information objects as well as their various attributes is informed by both cognitive and affective forces. These interactions will be presented more formally in the following model.

## 4. THE MODEL

Drawing from the user-side of Saracevic's relevance aspects, we present a model of subjective relevance using utility as the unifying concept. For the purpose of this paper we regard utility as being defined as including all expressions of user satisfaction and user-centered relevance derived from an interaction between the user and an information object, regardless of source or context. This concept includes pertinence, psychological relevance, situational relevance, usefulness, value and any other term used to qualify a user's cognitive or affective reactions to an information object, assuming of course that the interaction has some measurable effect on the user's sense of well being, happiness or satisfaction. This definition may seem broad, but since all the above concepts are equally unobservable and are based solely on user's preferences (by definition) no distinction need be made between definitions and criteria as to what specifically we *think* we are measuring. Basically, what the user sees as relevant *is relevant* [16]. Our work here is not concerned with whether or not any particular information object is relevant but with the various criteria users employ to make relevance judgments [15].

The internal and external aspects of relevance introduced in Section 3 are presented graphically in Figure 1. Cognitive and affective relevance both have a direct bearing on utility. Their interdependence is modeled by the two way arrow between them. To date there are no adequate measures for cognitive or affective relevance. For future studies we suggest the use of indexes made up of self-reported questions measured by Likert-type scales much like the measures used for utility. Understanding the various dimensions of cognitive and affective relevance and the various indicators of these dimensions (e.g., scale items) are important future questions.

Although situational relevance has no direct effect on utility, it does affect both cognitive and affective relevance. Situational effects must be processed internally in order to have an impact on subjective relevance. Since task has an effect on relevance we model situational relevance as having indirect effects on subjective relevance based on its direct effects on cognitive and affective relevance.

So the effects of cognitive and affective relevance on utility originate from three dimensions: (1) the user's *personal characteristics* such as personal preferences, emotional nature and cognitive ability, (2) the effects that *situation and other external factors* have on his or her motivations, and internal states and (3) the *interaction* between these two states. These are all illustrated in Figure 1 below.

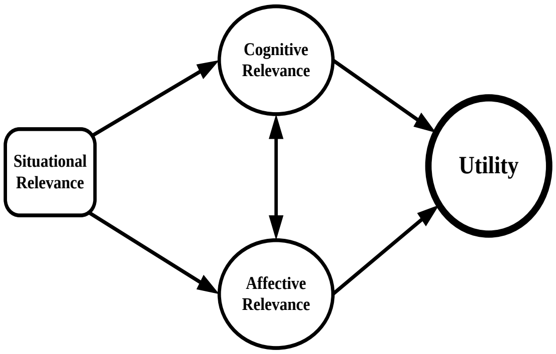


**Fig. 1.** Interactions of subjective relevance aspects.

## 4.1 Mathematical Model

In this section we present a mathematical representation of the constructs and interactions described in the theory section and illustrated in Figure 1. We use a utility function to represent the relationship between subjective relevance and its various aspects. We start with a general equation with no functional form.

$$\mathit{Utility(cog, aff, sit) = f(cog, aff, sit)} \qquad (1)$$

A user's utility of an information object is a function of cognitive relevance, affective relevance and situational relevance, as we spoke of above. Equation 1 serves a framework from which to proceed. It states the concept we wish to model, in this case utility. It also lists what is hypothesized as the determinants of utility, in this case cognitive, affective and situation relevance.

Since situational relevance has no direct effect on utility but does have direct effects on both affective and cognitive relevance we extract it from Equation 1 and express cognitive and affective relevance, and through them utility, as being conditional on situational relevance.

$$\mathit{Utility(cog, aff \mid T_i) = f(cog, aff \mid T_i) \quad i=0,1} \qquad (2)$$

As situational relevance changes, cognitive and affective relevances change as a result. These, in turn, have an effect on utility, in part because of the variation in task. In effect, task moderates the effects of affective and cognitive relevance.

The categorization of tasks depends on the type of study being performed. Tasks can be categorized by the user in naturalistic studies, or they can be imposed by the research in controlled experiments. Classification schemes vary but all represent the states of situational relevance that the researcher is interested in analyzing. The effects of task on cognitive and affective aspects of relevance are a ripe area for future research.

Equation 2 is analogous to having a separate utility function for each task class. Task effects on utility can be measured by the differences in mean utility across each task groups. More importantly, with adequate measures of affective and cognitive relevance, the means of these constructs can be compared across task groups as well.

At this point our model does not describe the relationships or interactions between and among these constructs; to do so we need to impose a functional form.

$$\mathit{Utility(cog, aff \mid T_i) = (\beta_0 + \beta_c cog + \beta_a\, aff + \beta_{ca}\, cog{*}aff) \mid T_i} \qquad (3)$$

There are a few points of interest with Equation 3. First, its functional form is quadratic, due to the interaction between *cog* and *aff*. Secondly, although it is not explicitly stated in the equation, task effects are still implicitly affecting cognitive and affective relevance.

As pointed out in our model and earlier, measurement is a huge issue in relevance research. Our theory depends on the development of adequate measure for testing of hypotheses and for the discovery of new aspects and dimensions of interest.

## 4.2 Comparative Statics

In economics, comparative statics is used to develop testable hypotheses from theoretical models [6]. It is used to analyze the effect that a change in one variable has on another. In utility theory it is used to measure the effects of changes in prices and income [13,17]. Prices and income are external constraints which are generally called *exogenous variables*. They are determined outside the model. In an experimental setting the researcher usually has some control over the exogenous variables, holding some constant and manipulating others. Task is the exogenous variable in our model. It is the external condition.

*Endogenous variables* are those that are determined inside the model. In the typical utility model quantities of goods are the endogenous variable. How much agents buy or consume is determined in part by the exogenous variables, price and income. Since utility is dependent on these quantities it is also an endogenous variable.

Utility is the variable of interest in both the classic consumer model and our model of subjective relevance. Here we use comparative statics to examine the change in utility (subjective relevance) as the result of a change in task. The effects that task have on cognitive and affective relevances are an intermediate step.

Now assume the situation changes from $T_0$ to $T_1$ where $T_0$ represents a task with high affective content and $T_1$ represents a task which has more cognitive demands associated with it. We further assume that the effect that $T_1$ has on affective relevance remains constant. So we have:

$$\frac{\delta cog}{\delta T} > \frac{\delta aff}{\delta T} > 0. \qquad (4)$$

Now the user balances the marginal benefits from using cognitive relevance criteria and using affective relevance criteria to the point where those marginal benefits are equal to each other and equal to zero. In other words the user cannot make any changes which will yield a better relevance decision according to his or her subjective criteria. This is called the marginal rate of substitution [13, 17]. Mathematically it is expressed as:

$$\frac{\delta U}{\delta cog} = \frac{\delta U}{\delta aff} \geq 0. \qquad (5)$$

So if cognitive influences on the user's utility goes up than the affective influences must go down to maintain the equilibrium in Equation 5.

## 5. CONCLUSION

In this paper we present a theory of subjective relevance. We use utility to conceptually model subjective relevance. We incorporate two dimensions (internal and external) and three aspects (cognitive, affective, situational) into our model. We then present a mathematical model and use comparative statics to illustrate the model.

Of course this model has short comings. Although it is intuitive that all subjective relevance comes from a combination of cognitive and affective factors, it is difficult to fully define these terms much less measure their effects. Still, thinking about subjective relevance along the lines proposed in this paper has advantages. The first is that no competing theory has yet established itself [15]. Secondly, its framework provides direction in relevance measurement research and creates a structural framework for testing hypotheses about internal, external, and interactive elements of subjective relevance.

One other shortcoming of our model is the exclusion of system-centered relevance from our analysis. The system side and user side also interact with each other. This interaction is hard to analyze. Having a relatively good understanding of the system side, research on the user side will likely bring a huge benefit and move toward an integrated understanding of the two.

Although utility is an excellent surrogate for subjective relevance, the other latent variables are more difficult to measure. If one topic of research is to follow from the study, developing appropriate measures for the cognitive and affective aspects of relevance would make the largest contribution. Cognitive and affective relevances exist only as *internal* processes. Situational relevance, however, consist of preferences over *external* objects. Assuming that the user strives to maximize his or her satisfaction or well-being over observable choices and self reported preferences, utility makes a good measure.

As much work has been done on task effects on relevance this still remains an important component of relevance research. Task is a variable in the system which can be easily manipulated thereby allowing for experiments and subsequent testing. It is also the only variable we consider as representing situational relevance. There are, of course, other such variables, but the growing body of research on task makes it a good candidate with which to start.

Having established a theoretical framework for subjective relevance, our future research will be to develop instruments for measuring cognitive, affective and situational aspects, and conduct an experiment to estimate the effects of this these constructs and the relationships among them.

## 6. REFERENCES

[1] Barry, C. 1994. User-defined Relevance Criteria: An Exploratory Study. *Journal of the American Society for Information Science, 45,* 149-159.

[2] Bentham, J. 1781. *An Introduction to the Principles of Morals and Legislation.* Retrieved from http://www.utilitarianism.com/jeremy-bentham/index.html.

[3] Borlund, P. 2003. The concept of relevance in IR. *Journal of the American Society for Information Science and Technology, 54*(10), 913-925.

[4] Cooper, W. S. 1973a. On Selecting a Measure of Retrieval Effectiveness, Part 1: The “Subjective” Philosophy of Evaluation. *Journal of the American Society for Information Science, 24,* 87-100.

[5] Cooper, W. S. 1973b. On Selecting a Measure of Retrieval Effectiveness, Part 2: Implementation of the philosophy. *Journal of the American Society for Information Science, 24,* 413-431.

[6] Currier, K.M. 2000. *Comparative Statics analysis in Economics.* World Scientific: River Edge, N.J.

[7] Happiness. In *Oxford English Dictionary* online. Retrieved April 7, 2009 from http://dictionary.oed.com

[8] Harter, S. 1992. Psychological relevance and information science. *Journal of the American Society for Information Science.* 49(3), 602-615.

[9] Ingwersen, P. & Järvelin, K. 2005. *The Turn: Integration of Information Seeking and Retrieval in Context.* Springer, Dordrecht.

[10] Kelly, D. 2007. Web page relevance: What are we measuring? *Workshop on Web Information Seeking and Interaction at the 30th Annual ACM International Conference on Research and Development in Information Retrieval (SIGIR '07),* Amsterdam, The Netherlands.

[11] Kelly, D. & Belkin, N.J. 2004. Display time as implicit feedback: Understanding task effects. In *Proceedings of the 27th Annual ACM International Conference on Research and Development in Information Retrieval (SIGIR '04),* Sheffield, UK, 377-384.

[12] Lopatovska, I., & Mokros, H. B. 2008. Willingness to pay and experienced utility as measures of affective value of information objects: Users’ accounts. *Information Processing & Management, 44*(1), 92-104.

[13] Mas-Colell, A., Whinston, M. D., & Green, J. R. (1995). *Microeconomic Theory.* Oxford University Press, New York.

[14] Saracevic, T. 1996. Relevance Reconsidered. In P. Ingwersen & N. O. Pors (Eds.), *Integration in Perspective. Proceedings of the Second International Conference on Conception of Library & Information Science (CoLIS 2),* Denmark, 201--218.

[15] Saracevic, T. 2007. Relevance: A Review of the Literature and a Framework for Thinking on the Notion in Information Science. Part II: Nature and Manifestations of Relevance. *Journal of the American Society for Information Science and Technology, 58*(13), 1915-1933.

[16] Swanson D. 1986. Subjective Versus Objective Relevance in Bibliographic Retrieval Systems. *The Library Quarterly, 56*(4), 389-398.

[17] Varian, H.R. 1978. *Microeconomic Analysis.* W.W. Norton & Co., New York.

[18] Wilson, P. 1973. Situational Relevance. *Information Storage & Retrieval, 9,* 457-469

[19] White, R.W. & Kelly, D. 2006. A study on the effects of personalization and task information on implicit feedback performance. *Proceedings of the 15th ACM international conference on Information and knowledge management, (CIKM '06),* 297-306.

# Query Reuse in Exploratory Search Tasks

Chirag Shah and Gary Marchionini
School of Information & Library Science
University of North Carolina
Chapel Hill, NC 27599-3360
chirag@unc.edu, march@ils.unc.edu

## ABSTRACT

In this paper, we present a number of observations and analyses from a user study. The study involved 84 subjects working on two different exploratory tasks for two sessions, which were one to two weeks apart. We found that a large portion of queries consisted of repetition of previously used query by the same user. There was also a high amount of overlap among the queries of different users for a given task, thus confirming the assumption that people tend to express their information request in the same/similar way for the same information need.

## Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *Query formulation, Search process.*

## General Terms

Design, Experimentation, Human Factors.

## Keywords

Exploratory search tasks, query re-usage.

## 1. INTRODUCTION

Exploratory searches typically exhibit a wide range of queries that in many cases take place over multiple sessions (White & Roth, 2009). As people search over time, they often reuse the same query either consciously or not. This phenomenon is pervasive, as illustrated by Teevan (2007) who found that one-third of all queries received by a search engine have been posed by the same user at least once before. Likewise, different people often use the same queries for the same or similar needs, a basis for several recommendation system techniques (e.g., Smyth, 2003). Understanding how and why people pose the same queries and how queries overlap across people are important problems theoretically and can also be used in practical ways to improve results for the individuals who executed the query or others who execute similar queries.

In this paper, we are present results from a study of how people use and reuse queries in multisession exploratory search tasks and to what extent these queries overlap across people and sessions.

Our analysis show that (1) there is a number of queries that people tend to reuse, and that (2) for the same information need, people tend to use same/similar queries.

## 2. METHOD

We were interested in looking at how people work in collaboration while performing an exploratory search task. We brought 42 pairs of people (total 84 subjects) to the lab for two sessions, which were separated by one to two weeks. The subjects were university students and staff from age 17 to 50. Pairs were recruited under the condition that they had previously done some collaborative project(s) together. Participants worked in different rooms so they cannot see or talk to each other directly. They were given a chat client for communication while working on the assigned tasks. For their first session participants completed some demographic questions and had a short practice session to familiarize themselves with the chat setup, which worked as a browser plug-in. They were asked to work through a task and after 20 minutes they were then interrupted, asked to complete a short questionnaire and asked to switch to the second task. They worked for about 20 minutes on the second task, completed another questionnaire and ended the session. Sessions lasted 75-90 minutes, which included a number of personal questionnaires and a group interview.

For the second session, the subjects were asked to resume their tasks from the previous session to collect more relevant information and summarize their findings. They worked on each of the tasks for about 20 minutes, including creating their summaries.

The two tasks given to the subjects are listed below.

Task-1: Economic recession

*"A leading newspaper has hired your team to create a comprehensive report on the causes and consequences of the current economic recession in the US. As a part of your contract, you are required to collect all the relevant information from any available online sources that you can find.*

*To prepare this report, search and visit any website that you want and look for specific aspects as given in the guideline below. As you find useful information, highlight and save relevant snippets. Later, you can use these snippets to compile your report. You may also want to save the relevant websites as bookmarks, but remember - your main objective here is to collect as many relevant snippets as possible.*

*Your report on this topic should address the following issues: reasons behind this recession, effects on some major areas, such as health-care, home ownership, and financial sector (stock market), unemployment statistics over a period of time, proposal,*

*HCIR 2009*, October 23, 2009, Washington DC, USA.

*execution, and effects of the economy stimulation plan, and people's opinions and reactions on economy's downfall."*

Task-2: Social networking

*"The College Network News Channel wants to do a documentary on the effects of social networking services and software. Your team is responsible for collecting various relevant information (including statistics) from the Web. As a part of your assignment, you are required to collect all the relevant information from any available online sources that you can find.*

*To prepare this report, search and visit any website that you want and look for specific aspects as given in the guideline below. As you find useful information, highlight and save relevant snippets. Later, you can use these snippets to compile your report. You may also want to save the relevant websites as bookmarks, but remember - your main objective here is to collect as many relevant snippets as possible.*

*Your report on this topic should address the following issues: emergence and spread of social networking sites, such as MySpace, Facebook, Twitter, and del.icio.us, statistics about popularity of such sites (How many users? How much time they spend? How much content?), impacts on students and professionals, commerce around these sites (How do they make money? How do users use them to make money?), and examples of usage of such services in various domains, such as health-care and politics."*

While the experiment was designed to study how people seek information in collaboration, for the purpose of this paper, we will consider an individual subject as a unit.

## 3. OBSERVATIONS AND ANALYSES

In this section we present several observations and analyses of how individuals use and re-use queries, and how individual queries overlap with those of the other people with the same information need.

### 3.1 Query usage

We first consider overall query usage and re-usage. Our subjects used total 4207 queries (aggregated over both the tasks and sessions), of which 1605 were source-wise unique queries, and 1522 were overall unique queries. Thus, nearly 40% of the queries issued were repeated at least once (by the same or some other subjects). Figure 1 shows the sources that were queried by the participants in aggregate with source-wise number of queries (total and unique). A significant portion of all the queries was sent to Google, with CNN (mostly for Task-1) and Bing next most used. It is interesting to note that with every source, a large portion of queries were repeats.

Figure 2 lists the top sites that the users visited for each task. Google was the most visited site for both the tasks, followed by news sites, such as the New York Times and CNN. For Task-1, the subjects also visited Bureau of Labor Statistics (BLS) and Recession.org website, where many up-to-date statistics on the current economic recession can be obtained. We found that these websites were discovered mostly due to queries such as "unemployment statistics US" and "economic recession" rather than by following links with sites or directly typing in URLs.[1]

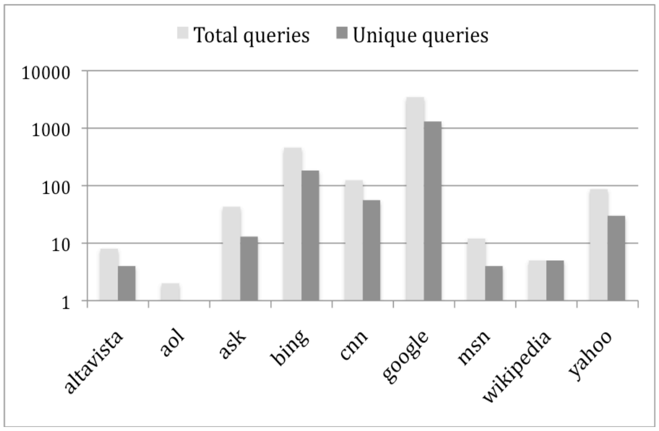
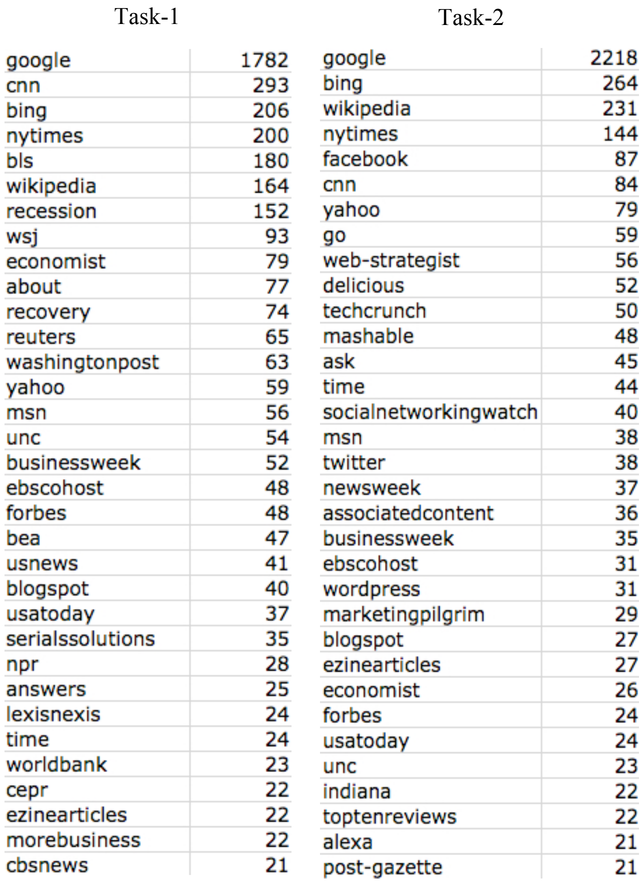


**Figure 1: Source-wise query usage and overlap in aggregation (number of queries in log scale)**



| Task-1 | | Task-2 | |
|---|---|---|---|
| google | 1782 | google | 2218 |
| cnn | 293 | bing | 264 |
| bing | 206 | wikipedia | 231 |
| nytimes | 200 | nytimes | 144 |
| bls | 180 | facebook | 87 |
| wikipedia | 164 | cnn | 84 |
| recession | 152 | yahoo | 79 |
| wsj | 93 | go | 59 |
| economist | 79 | web-strategist | 56 |
| about | 77 | delicious | 52 |
| recovery | 74 | techcrunch | 50 |
| reuters | 65 | mashable | 48 |
| washingtonpost | 63 | ask | 45 |
| yahoo | 59 | time | 44 |
| msn | 56 | socialnetworkingwatch | 40 |
| unc | 54 | msn | 38 |
| businessweek | 52 | twitter | 38 |
| ebscohost | 48 | newsweek | 37 |
| forbes | 48 | associatedcontent | 36 |
| bea | 47 | businessweek | 35 |
| usnews | 41 | ebscohost | 31 |
| blogspot | 40 | wordpress | 31 |
| usatoday | 37 | marketingpilgrim | 29 |
| serialssolutions | 35 | blogspot | 27 |
| npr | 28 | ezinearticles | 27 |
| answers | 25 | economist | 26 |
| lexisnexis | 24 | forbes | 24 |
| time | 24 | usatoday | 24 |
| worldbank | 23 | unc | 23 |
| cepr | 22 | indiana | 22 |
| ezinearticles | 22 | toptenreviews | 22 |
| morebusiness | 22 | alexa | 21 |
| cbsnews | 21 | post-gazette | 21 |

**Figure 2: Top sites visited for both the tasks**

[1] A few times subjects even typed queries such as "bls" in Google.

## 3.2 Query re-usage and overlap

We were interested in studying four particular questions about query re-usage (within individuals) and overlap (between individuals). These questions and the corresponding observations and analyses follow.

**Q1. How often do people re-use their own queries?**

Figure 3 plots the total number of queries used for each task and session, along with how many of them were unique. As we can see, a large portion of queries for a given session was already used in that session (about 20 minutes in length). This is also reflected in Table 1, where the query re-use statistics are reported. Both the tasks had on average about 40% of query re-use.

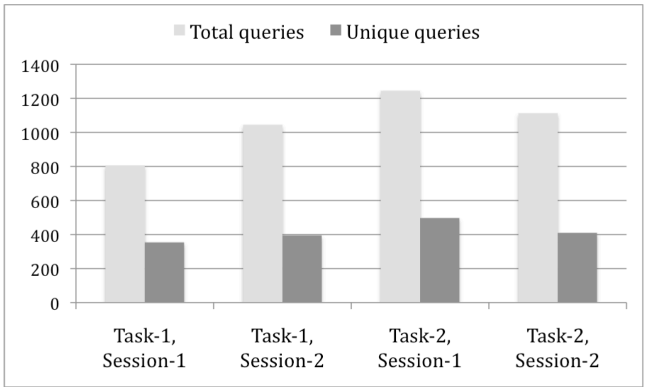


**Figure 3: Task and session-wise individual query re-usage**

**Table 1: Task and session-wise individual query re-usage statistics**

| | Session-1 | Session-2 | Average |
|---|---|---|---|
| **Task-1** | 44.02% | 37.79% | **40.91%** |
| **Task-2** | 39.92% | 36.84% | **38.38%** |
| **Average** | **41.97%** | **37.32%** | |

We also looked at the differences in query re-usage behavior for each user across tasks and sessions. In order to do this, we compared a user's query re-usage for a given task and session to other task and/or session. We then did a pair-wise comparison for all 84 subjects for their tasks and sessions. The significance test results for these comparisons are provided in Table 2. As we can see, for Task-2, there was a statistically significant difference between how people re-used their queries between two sessions. We also found statistically significant difference in query re-usage between the second sessions for both the tasks. Other comparisons showed non-significant differences in individual query re-usage behavior.

**Q2. How often people re-use their queries from the previous session?**

One of our interests in this study was to look at browsing and query re-use across multiple sessions. We found (Figure 4) that only about 5-10% of the queries used in the second session were repeats from the first session. However, when we expanded our matching criteria to include subqueries (e.g., "economics recession" is a subquery of "economics recession US"), we found a much larger re-usage portion. In fact, for Task-2, we found nearly half of the queries being repeats (as the same exact query or a subquery) from the first session. Our analysis showed that for Task-2, more than half of the queries had "social networking" in them. This may be due to the fact that almost all the facets in this task also had "social networking" as a sub-facet; the subjects found it difficult to investigate those facets without the context of social networking. This also became apparent in the interviews we did after the tasks. For Task-1, on the other hand, the subjects could run fairly independent queries for covering different facets, such as "unemployment stats" and "recession causes".

**Table 2. Two-tailed paired *t*-test for measuring the significance of difference in query reusing for different tasks and sessions. Statistical significance at *p<0.05* is show in bold.**

| Comparison | Value of *p* |
|---|---|
| Task-1: Session-1 to Task-1: Session-2 | 0.877 |
| Task-1: Session-1 to Task-2: Session-1 | 0.506 |
| Task-2: Session-1 to Task-2: Session-2 | **0.000** |
| Task-1: Session-2 to Task-2: Session-2 | **0.002** |

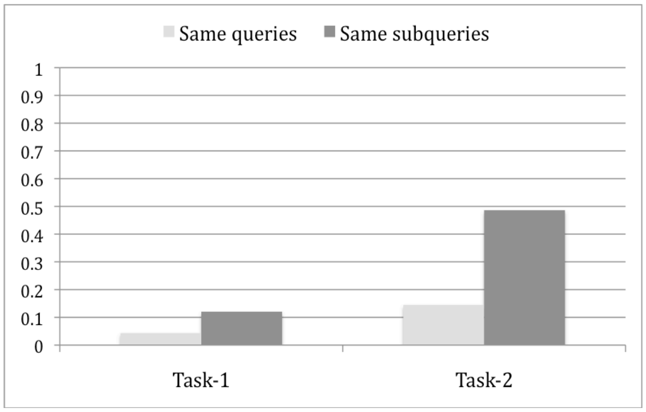


**Figure 4: Second session query and subquery re-usage rates from the first session**

**Q3. What proportion of queries overlap across different people for the same task?**

Figure 5 shows the portion of queries that one used was also used by some other subject for the same task. Similar to above, we looked at not only exact query match, but also subquery matches. We can see that for Task-2, the subjects had a much better agreement on what the queries were reused. This confirms our justification given at the end of the previous question analysis.

**Q4. What proportion of queries is similar across different people of different teams for the same task?**

Instead of simply looking at exact matching queries, we also looked at how close two given queries are. To find this closeness, we used Edit Distance measure. The results are plotted in Figure 6 and 7. In these figures, the X-axis shows the edit distance between two queries, and Y-axis shows the number of queries. Thus, for Task-1, there were 528 queries that had a closest query with distance zero (exact match), 162 queries that had a closest query with distance one, and so on. For simplicity, edit distance only up to 20 is shown in both the graphs.

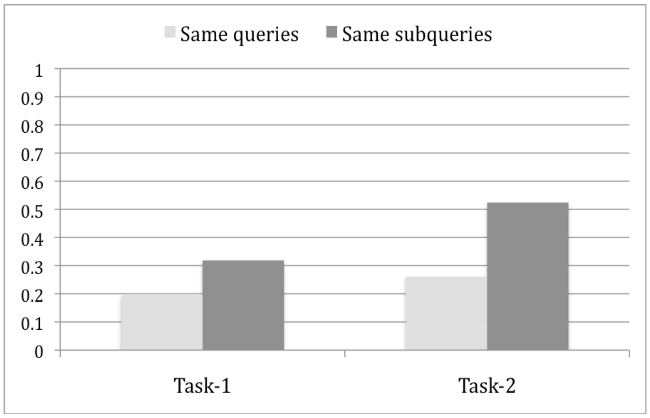


**Figure 5: Average query and subquery re-usage proportion for a subject with respect to other subjects**

Once again, we find that many queries that our subjects used for a given task were the same or very similar to the queries other(s) have used.

We also found that in case of Task-2, there was a greater agreement among the participants in formulating the queries, as compared to Task-1 (e.g., 725 queries with zero edit distance for Task-2 vs. 528 for Task-1).

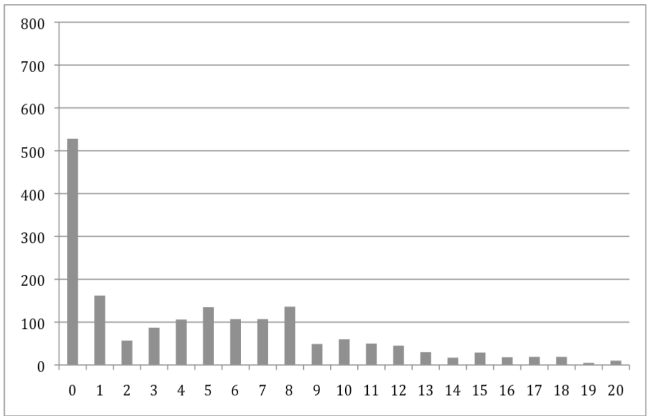


**Figure 6: Edit Distance among the queries for Task-1. X-axis shows Edit Distance between a pair of queries, and Y-axis shows number of queries.**

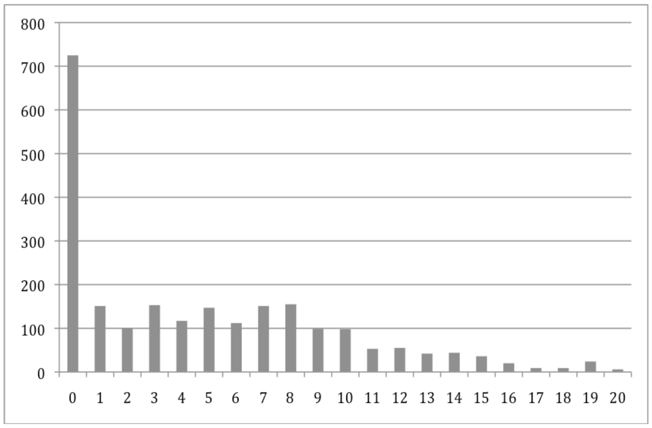


**Figure 7: Edit Distance among the queries for Task-2. X-axis shows Edit Distance between a pair of queries, and Y-axis shows number of queries.**

## 4. DISCUSSION

From an analysis of query usage of 84 subjects working on exploratory tasks over two sessions, we found support for query re-usage for individuals, and high overlap among the queries of multiple subjects for a given task. Such observations and analyses about query usage and re-usage confirm is that people with same information need tend to express their information need in the same/similar way. This is a driving motivation for collaborative filtering work and query assistance/suggestion. The substantial reuse and overlap demonstrate that such techniques may be even more useful for exploratory searching.

## 5. REFERENCES

[1] Smyth, B., Balfe, E., Briggs, P., Coyle, M., Freyne, J. (2003). Collaborative Web Search. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, p. 1417-1419. August 9-15, 2003. Acapulco, Mexico.

[2] Teevan, J. 2007. The re:search engine: simultaneous support for finding and re-finding. In *Proceedings of the 20th Annual ACM Symposium on User interface Software and Technology* (Newport, Rhode Island, USA, October 07 - 10, 2007). UIST '07. ACM, New York, NY, 23-32.

[3] White, R. & Roth, R. (2009). *Exploratory search; Beyond the query-response paradigm*. San Franscisco: Morgan-Claypool.

# Towards Timed Predictions of Human Performance for Interactive Information Retrieval Evaluation

Mark D. Smucker
Department of Management Sciences
University of Waterloo
msmucker@uwaterloo.ca

## ABSTRACT

Today's popular retrieval metrics are largely divorced from any notion of a user interface or a user model. These metrics such as mean average precision produce measures of ranked results quality rather than predictions of human performance. Using GOMS, we modify the Cranfield-style of evaluation to create a new evaluation method that makes testable predictions of human performance. While not yet validated by user studies, we demonstrate using our evaluation method that such an evaluation technique gives information retrieval researchers the ability to understand how changes in the interface or in the underlying retrieval algorithm impact user performance. Future work should be directed to the creation and validation of evaluation methods that predict user performance and incorporate explicit user interfaces and user models.

## 1. INTRODUCTION

While the information retrieval (IR) community has known since the work of Dunlop [8] that IR evaluation could be improved with automated usability methods from the field of human computer interaction (HCI) [9], retrieval metrics devoid of explicit user interfaces and user models continue to dominate IR evaluation.

As a step towards answering our call to move Cranfield-style evaluation towards a more realistic evaluation [14], we use GOMS, an automated usability method, to create an evaluation method that makes testable predictions of human performance. Cranfield-style evaluation measures the ranking quality of a retrieval algorithm given a test collection of documents, search topics, and relevance judgments. An example of a commonly used metric in Cranfield-style evaluation is precision at rank 10 (P10). The precision at rank 10 is equal to number of relevant documents found within the first 10 documents returned by a retrieval algorithm divided by 10. While these metrics can be somewhat correlated with user performance [1, 2, 15] they do not make testable predictions of user performance.

GOMS estimates the time for expert users to complete a task given a certain interface [4, 10]. The acronym GOMS stands for Goals, Operators, Methods, and Selections. In simple terms, GOMS is about finding the sequence of operations on a user interface that allows the user to achieve the user's goal in the shortest amount of time. GOMS allows an interface designer to obtain predicted task times for different interfaces before more expensive user testing.

In our case, the IR user has a goal of finding as many relevant documents as possible. The operations are the actions possible with a hypothetical user interface. We embody "methods and selections" in what we refer to as a *user model*. The user model we create in this paper is a simple, first step towards better models. For example, our model lacks the ability to perform query reformulation. Considerable research effort will be required to create user models based on observed user behavior. Even with a better user model, the overall evaluation methodology will need to be validated with user studies to determine the accuracy of the performance predictions [5, 6].

By combining GOMS with the Cranfield-style of evaluation, we obtain a simulation of user behavior for which all user actions have associated times. For example, from GOMS we know that moving the mouse to a button will take on average 1.1 seconds [11]. From this simulation many testable measures of human performance are computable. In this paper, we compute the number of relevant documents read by the simulated user within 10 minutes.

Evaluation methods, such as ours, that explicitly incorporate a user interface and a user model allow IR researchers to investigate the impact of interface changes on user performance before turning to more expensive user studies for confirmation. In other words, IR researchers can simulate user behavior over a hypothetical user interface to generate testable hypotheses. For example, based on our experimental results, we hypothesize that the user interface determines the relationship between ranked retrieval quality and user performance.

Next we describe our method in more detail and then follow with our experiments and preliminary results.

## 2. METHODS AND MATERIALS

Our evaluation methodology consists of a hypothetical user interface and a user model defined over that interface. Our hypothetical interface is a simplified version of today's common web search interface. The interface provides a text box that allows the user to enter and submit a keyword-like query. On submission of the query, the user is presented with 10 query-biased summaries of the top ranked results produced by an underlying retrieval algorithm in response to the query. Each result summary provides a hyperlink or button that when clicked on will take the user to the full document. The interface provides the means for the user to hit a "back button" and return to the search results. The

*Third Workshop on Human-Computer Interaction and Information Retrieval (HCIR'09),* October 23, 2009, Washington DC, USA.

```
Let $t$ be the total search time.
Enter query and hit return. ($t \leftarrow t + K(\text{length(query)}+1)$)
Wait for results & move hands to mouse. ($t \leftarrow t + W$)
for $i \leftarrow 1$ to Number of Results do
    Read and evaluate summary. ($t \leftarrow t + SE$)
    $D \leftarrow$ document at result $i$
    judgment $\leftarrow$ qrels judgment of $D$
    if judgment is non-relevant then
        With probability $P0$ decide to read $D$.
    else if judgment is relevant then
        With probability $P1$ decide to read $D$.
    else // judgment is highly relevant
        With probability $P2$ decide to read $D$.
    end if
    if Decided to read $D$ then
        Point mouse to link/button. ($t \leftarrow t + P$)
        Click mouse button. ($t \leftarrow t + BB$)
        Wait for result page to load. ($t \leftarrow t + W$)
        Read and evaluate $D$. ($t \leftarrow t + DE$)
        if judgment is relevant or judgment is highly relevant then
            numRelevantRead $\leftarrow$ numRelevantRead + 1
        end if
        Point mouse to back button. ($t \leftarrow t + P$)
        Click mouse button. ($t \leftarrow t + BB$)
    end if
    if ((i+1) mod 10) = 1 then // Only 10 results per page
        Point mouse to next page link/button. ($t \leftarrow t + P$)
        Click mouse button. ($t \leftarrow t + BB$)
        Wait for next page of results. ($t \leftarrow t + W$)
    end if
end for
```

**Figure 1: User model. The time each action takes is shown in parentheses. Table 1 lists the model parameters and their values.**

| | |
|---|---|
| Keystroke (average non-secretarial typist 40 wpm) [11] | $K = 0.28$ s |
| Type a sequence of $n$ keys [11] | $n \times K$ s |
| Point the mouse to a target on the display [11] | $P = 1.1$ s |
| Press or release the mouse button [11] | $B = 0.1$ s |
| Click mouse button (press and release) [11] | $BB = 0.2$ s |
| Move hands to keyboard or mouse [11] | $H = 0.4$ s |
| Mental act of routine thinking or perception [11] | $M = 1.2$ s |
| Wait for search results or web page to load | $W = 1$ s |
| Time to evaluate a search result summary [16] | $SE = 19$ s |
| Time to evaluate a document for relevance [16] | $DE = 88$ s |
| Probability of clicking on non-relevant summary [16] | $P0 = 0.25$ |
| Probability of clicking on relevant summary [16] | $P1 = 0.53$ |
| Probability of clicking on highly relevant summary [16] | $P2 = 0.77$ |

**Table 1: User model parameters. All times are in seconds. Figure 1 shows the user model.**

search results interface also provides a link or button to take the user to a new page with the next 10 ranked results.

Figure 1 shows our user model. First, the simulated user enters the query by typing and then waits for the first 10 search results. The user then proceeds to read and evaluate the result summaries one after the other. With some probability conditional on the relevance of the underlying document, the user will decide to click on a summary and read the document. After reading the document, the user hits a "back button" and continues reading and evaluating the search result summaries. When the user reaches the end of the summaries on a page, the user clicks on a link or button to request the next 10 results. All actions have associated times.

Our user model is simple, for demonstration purposes, and not an attempt to capture the complex process of search. For example, while query reformulation could be made possible with our hypothetical interface, our user model is incapable of reformulating queries. Eye tracking research has clearly shown that users quickly reformulate queries that don't produce top ranked relevant documents [12].

Table 1 lists the parameter settings of our user model. These settings come primarily from two places. For GOMS, we utilize the keystroke level model (KLM) [3]. In this model, the operators are defined at the level of keystrokes and mouse movements. Timings for these operators are averages obtained from various user studies [11]. In our use of GOMS, we inadvertently omitted use of the "mental" operator. Even so, most mental actions in our model are involved in the evaluation of the search result summaries and documents and are captured by the $SE$ and $DE$ parameters.

Our other source for parameter settings comes from the work of Turpin, Scholer, Järvelin, Wi and Culpepper [16] who created a methodology to include search result summaries into standard list quality metrics such as precision at 10 (P10) and mean average precision (MAP). As part of their work, they asked users to determine whether or not to click on a summary and view the corresponding document. If the user felt the summary would lead to a relevant document, the user would decide to click on the summary. Users then judged the relevance of documents on a 4 point graded scale. On average, users took 19 seconds to evaluate a summary and 88 seconds to evaluate a document. While we know that eye tracking results show that users usually spend much less than 19 seconds reading a summary [7], we utilize Turpin et al.'s timings to be consistent with their measures of summary evaluation accuracy.

In a simulation analysis of TREC 9 and 10 submitted runs, Turpin et al. mapped their two highest relevance categories to TREC's "highly relevant" and their least relevant category to "relevant" and finally mapped non-relevant to non-relevant. With this mapping, the probability that a user would click on a summary was 0.77 for highly relevant documents, 0.53 for relevant, and 0.25 for non-relevant. These summary evaluation accuracies are in line with the 75% accuracy found by Sanderson [13]. We use these probabilities in our experiments with the same TREC 9 runs.

## 3. EXPERIMENTS

For our experiments we use the 40 automatic, title only ad-hoc web retrieval runs from TREC 9. For each run we compute the precision at 10 (P10) as well as the number

**Average Improvement over Normal Model**

| Condition | Percent Imp. |
|---|---|
| Perfect Summaries | 80% |
| Read Documents Twice as Fast | 38% |
| Better Summaries | 23% |
| Read Summaries Twice as Fast | 17% |

**Table 2: Results for the 4 interface improvements described in Section 3. For each of the 40 TREC-9 runs, user performance is measured as the number of relevant documents read within 10 minutes.**

of relevant documents read by our simulated user within 10 minutes. Because there is inherent randomness in our user model caused by the different probabilities of clicking on a result summary, we simulate usage 1000 times for each topic of each run and average the predicted performance.

In addition, we examine 4 possible interface improvements:

1. We modify the result summaries so that users can evaluate them twice as fast (9.5 s rather than 19 s).

2. We improve the evaluation accuracy of summaries. For relevant and highly relevant documents, the summary evaluation accuracy increases by 25% (0.53 to 0.663 and 0.77 to 0.963) and for non-relevant documents the error rate decreases by 25% (0.25 to 0.188).

3. We provide some means for the users to evaluate documents twice as fast (44 s rather than 88 s).

4. We make summaries perfect. All relevant and highly relevant documents are viewed, and users waste no time reading non-relevant documents. While likely an impossible interface improvement if evaluation time remains unchanged, this change allows us to see the maximum possible gain for improvements in summary evaluation accuracy.

We naively assume all interface improvements do not affect other aspects of the search process. For example, for improvement 1 above, users can evaluate summaries faster with no decrease in evaluation accuracy.

## 4. RESULTS AND DISCUSSION

Table 2 shows that our evaluation method predicts that each of the interface improvements would increase the number of relevant documents evaluated by the user within 10 minutes.

While our interface improvements are all "what-if" experiments, we can see in Figure 2 that under the assumptions of our evaluation method, the user interface determines the relationship between ranked retrieval quality and user performance. What good is a 20% improvement in P10? The answer depends on the quality of the user interface. Better interfaces better translate retrieval gains into user performance gains.

Based on Figure 2, should we conclude that P10 is a metric that mirrors user performance when performance is defined to be the number of relevant documents examined within 10 minutes? No. We've replaced one evaluation method of retrieval quality with another but neither have been validated against actual human performance.

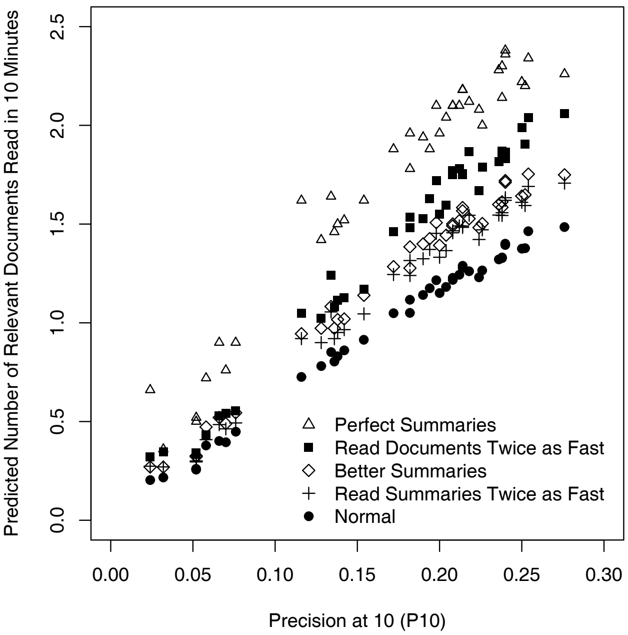


**Figure 2: This figures shows the precision at 10 (P10) vs. the predicted number of relevant documents read within 10 minutes for each of the 40 TREC 9 runs and the 5 interface conditions described in Sections 2 and 3.**

What we have with our new evaluation method is a method that aims to be directly predictive of the variable of concern: human performance. Precision at 10 or MAP does not attempt to predict human performance. P10 and MAP and metrics like them output a measure of list quality that is loosely coupled with user performance.

What a method like our simple example does is that it marries together retrieval quality, a user model, and the hypothetical user interface and makes a prediction concerning user performance. All of these 4 important parts of an evaluation of retrieval performance are explicit in our evaluation.

The significant shift in thinking that our evaluation method brings about is that when an evaluation method contains all of these components, we gain the ability to start asking questions about what will most improve human performance. In other words, we can look to see where the user is spending time. Is the most time spent manipulating the interface? Or is it spent wading through non-relevant documents? Or is it spent reading documents? Our evaluation method allows the IR researcher to gain insight to these questions.

## 5. CONCLUSION

We combined an automated usability method, GOMS, with the Cranfield-style of evaluation to produce a new evaluation method that produces testable predictions of human performance. This evaluation method allows IR researchers to investigate the impact of various interface improvements and also to see the degree to which changes in retrieval quality affect user performance. Future work remains to create accurate, predictive evaluation methods that explicitly incorporate both the user interface and a model of the user's search behavior.

## 6. ACKNOWLEDGMENTS

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), and in part by the University of Waterloo. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect those of the sponsors.

## 7. REFERENCES

[1] A. Al-Maskari, M. Sanderson, P. Clough, and E. Airio. The good and the bad system: does the test collection predict users' effectiveness? In *SIGIR'08*, pages 59–66. ACM, 2008.

[2] J. Allan, B. Carterette, and J. Lewis. When will information retrieval be "good enough"? In *SIGIR'05*, pages 433–440. ACM, 2005.

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

[4] S. K. Card, A. Newell, and T. P. Moran. *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates, Inc., Mahwah, NJ, USA, 1983.

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

[6] W. S. Cooper. On selecting a measure of retrieval effectiveness: Part ii. implementation of the philosophy. *JASIS*, 24(6):413–424, Nov/Dec 1973.

[7] E. Cutrell and Z. Guan. What are you looking for?: an eye-tracking study of information usage in web search. In *CHI'07*, pages 407–416. ACM, 2007.

[8] M. D. Dunlop. Time, relevance and interaction modelling for information retrieval. In *SIGIR'97*, pages 206–213. ACM, 1997.

[9] M. Y. Ivory and M. A. Hearst. The state of the art in automating usability evaluation of user interfaces. *ACM Computing Surveys*, 33(4):470–516, 2001.

[10] B. E. John and D. E. Kieras. Using GOMS for user interface design and evaluation: which technique? *ACM Transactions on Computer-Human Interaction*, 3(4):287–319, 1996.

[11] D. Kieras. Using the keystroke-level model to estimate execution times. `ftp://ftp.eecs.umich.edu/people/kieras/GOMS/KLM.pdf`, copy obtained via Google, `http://74.125.95.132/search?q=cache:wvKGAmd5KIIJ:ftp://ftp.eecs.umich.edu/people/kieras/GOMS/KLM.pdf`, 2001.

[12] L. Lorigo, M. Haridasan, H. Brynjarsdóttir, L. Xia, T. Joachims, G. Gay, L. Granka, F. Pellacini, and B. Pan. Eye tracking and online search: Lessons learned and challenges ahead. *JASIS*, 59(7):1041–1052, 2008.

[13] M. Sanderson. Accurate user directed summarization from existing tools. In *CIKM'98*, pages 45–51. ACM, 1998.

[14] M. D. Smucker. A plan for making information retrieval evaluation synonymous with human performance prediction. In *Proceedings of the SIGIR 2009 Workshop on the Future of IR Evaluation*, pages 11–12, July 2009.

[15] A. Turpin and F. Scholer. User performance versus precision measures for simple search tasks. In *SIGIR'06*, pages 11–18. ACM, 2006.

[16] A. Turpin, F. Scholer, K. Jarvelin, M. Wu, and J. S. Culpepper. Including summaries in system evaluation. In *SIGIR'09*, pages 508–515. ACM, 2009.

# The Information Availability Problem

Daniel Tunkelang
Endeca
dt@endeca.com

## ABSTRACT

In recent years, library and information scientists, particularly those concerned with interactive information retrieval, have complained that the information retrieval community--both researchers and practitioners--overemphasizes precision as a performance measure. More precisely, the IR community favors measures that emphasize precision in the top-ranked results, either explicitly (e.g., p@10) or implicitly (e.g., average precision, DCG). This essay advocates the study of the *information availability* problem, a general information seeking problem ill-served by today's models, evaluation measures, and tools. It defines the problem, proposes evaluation criteria for it, and explores how current and future tools could address it. Finally, it considers a testing approach based on the “games with a purpose” framework.

## Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *information filtering*; H.1.2 [**Models and Principles**]: User/Machine Systems – *human factors, human information processing*

## General Terms

Algorithms, Performance, Experimentation, Human Factors.

## Keywords

interactive information retrieval, recall, models, evaluation

## 1. INTRODUCTION

Tefko Saracevic has described the chasm between system-centered information retrieval researchers and user-centered library and information scientists as a "battle royal" [1], perhaps best summed up in the dialogue with which Rijsbergen opens his book on The Geometry of Information Retrieval [2]. The participants are "B" (Bruce Croft, representing the system-centered view), "N" (Nick Belkin, representing the user-centered view), and "K" (the author). While neither side has surrendered, both acknowledge that the system-centered approach controls far more territory, in terms of both research publications and influence on commercial implementations.

Moreover, it is not just any system-centered approach that has dominated, but specifically one that focuses on ranked retrieval and precision in the top-ranked results. Google's "I Feel Lucky" button reflects an endorsement of p@1 as a performance measure. In a less extreme form, the conventional wisdom is that users will quickly abandon a web site or application in frustration if they cannot resolve their information need using the first page of results returned by a search engine, i.e., the ten blue links.

## 2. RELATED WORK

Despite precision in the top-ranked results as a dominant performance measure, there is no lack of alternatives in the research literature. The following list makes no claim to be exhaustive (I leave that ambitious project to Stefano Mizzaro [3]!), but rather offers highlights representing different conceptions of retrieval performance.

**Recall**. Originally used as a set retrieval measure, recall largely shows up today as a component of the f1 measure (the harmonic mean of precision and recall) or in specialized domains like e-discovery. Interestingly, a recent essay by Zobel et al. questions whether recall makes sense as a measure even for those domains [4]. Moreover, research by Dostert and Kelly suggests that people are poor estimators of recall while pursuing recall-oriented search tasks [5].

**k-call**. Karger and Chen proposed k-call at n, a binary measure returns 1 if at least k of the top n results are relevant, 0 otherwise [6]. The k-call measure is similar to the **%no** measure proposed earlier by Ellen Voorhees [7].

**Uncertainty**. Kuhlthau [8] and others following her (e.g., Wilson et al. [9]) have looked at uncertainty as a holistic effectiveness measure for the information seeking process.

**PRP for IIR**. Fuhr proposed this framework to generalize the probability ranking principle, a classical basis for batch retrieval, to an interactive framework that more realistically models actual information seeking behavior [10].

**Findability**. Ma et al. propose a findability measure as how reliably players locate a page in a human computation game [12].

## 3. MODEL

The *information availability* problem represents an extreme case for which the importance of recall dominates that of precision. Its premise is that an information seeker faces uncertainty as to whether or not some specified information of interest is available through an information seeking support system. Instances of this problem include many high-value information tasks, such as those facing national security and legal/patent professionals, who spend hours or days trying to determine whether the desired information exists. In the problem’s most basic form, the information, if available, resides in a single document.

The information availability problem is a realistic use case for testing the effectiveness of information seeking support systems—particularly those that aim to support interaction and exploration. On one hand, the problem is sufficiently concrete to allow for quantitative assessment of user performance. On the other hand, the problem is inherently about task performance rather than query performance, and makes it possible to compare the effectiveness of different interface approaches, or of variations within the same interface.

## 4. EVALUATION

We propose the following three evaluation criteria:

**Task success**. If the information of interest is available, the user achieves *positive success* by discovering it. If not, then the user achieves *negative success* by correctly deducing that it is not available. Task failure occurs when the user gives up prematurely, even though the information is available.

**Efficiency**. The efficiency of the user is measured simply by the amount of time required to complete the task (successfully or not). Since the task is directed, the user is not expected to spend time in undirected exploration.

**Confidence**. When the user achieves positive success, there is no question as to the user's confidence in the result. The negative case, however, is another story. The user's confidence in a negative outcome could range from complete (but possibly misplaced) certainty to complete doubt, e.g., giving up out of frustration. The user's subjective (and self-reported) confidence is an important measure for the negative case.

## 5. CURRENT TOOLS

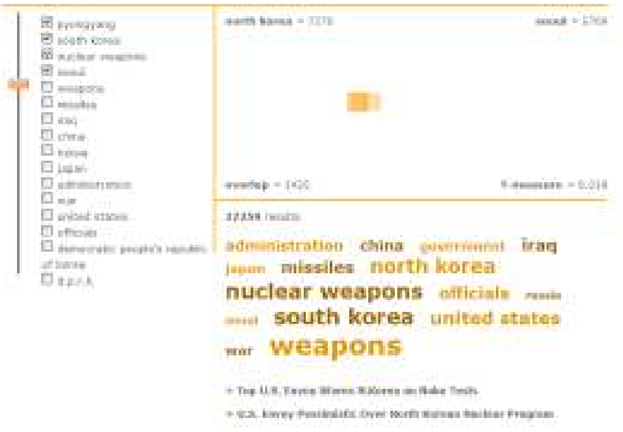
Before considering new tools to address the information availability problem, let us consider how existing tools address it.

First, there is the tool most commonly applied to information seeking today in general: ranked retrieval. For some instances of the information availability problem, ranked retrieval is quite effective—namely, the easy cases where the information is available and high precision in the top ranked results quickly leads to positive success. Unfortunately, ranked retrieval stumbles when the information need is difficult for a user to express unambiguously, particularly given the limitations of the user's knowledge of how the information in the system is represented and how the system processes queries [11]. More importantly, ranked retrieval breaks down completely in the negative case: users eventually get frustrated with reformulating their queries and then give up. It is not clear how users can calibrate their confidence in a negative outcome, other than by learning from their own experiences, i.e., extrapolating from instances where they later discover that they failed to find available information.

Then, there are tools that are explicitly designed to support exploration and interactive information retrieval. These, which include faceted search and query suggestion systems, seem more promising. In particular, approaches built on set retrieval rather than ranked retrieval are a better fit for a problem that emphasize recall rather than precision. These approaches, however, require sophisticated indexing of the content that may not always be practical. For example, faceted search requires that documents be associated with facet values—which in turn requires both a faceted classification scheme and a means of applying it to the corpus.

## 6. POTENTIAL NEW TOOLS

Two extensions to today's query suggestion systems could help address the information availability problem.



The first is the use of query suggestion to increase recall by broadening query results, rather than to increase precision by narrowing or re-ranking them.

The second is the incorporation of query previews into a query suggestion system in order to reduce the actual and perceived cost of exploration.

The figure above shows a prototype of a tool that incorporates these two elements. In the example interaction shown, a user has imitated a search against a news collection with the query *north korea*. The system produces a list of ranked query term suggestions: *pyonyang*, *south korea*, *nuclear weapons*, etc. The user can expand the query to include these terms either by selecting their corresponding checkboxes or by moving a slider down to include all of the terms up to that point in the list.

As the user manipulates the query expansion, the system offers instantaneous feedback showing how the change affects the query. The feedback includes three elements:

**Term relatedness**. For a given term that the user is considering adding to the query (here, *seoul*), the system shows statistics relating the term to original query (in this case, *north korea*).

**Example documents**. As the user expands the query, the system immediate shows example documents from the expanded result set. These example documents are newly introduced documents that are representative of the expanded set.

**Topic summary**. The tag cloud shows the topics most represented in the result set associated with the current query expansion. The instantaneous feedback allows the user to visualize topic drift and back off if the expansion takes the query on an unproductive tangent.

With respect to the information availability problem, such an approach complements techniques aimed at increasing query precision. We can imagine a two-phase approach. In the first phase, the user employs techniques like faceted search to progressively narrow a query—a query elaboration process aimed at precisely expressing the user's intent. In the second phase, the user employs a tool like that shown in the prototype to broaden from this precise query and thus expand recall. Ultimately, the goal is for the user to achieve high recall for his or her information need, and thus to either efficiently achieve positive or negative success.

## 7. GAMES WITH A PURPOSE

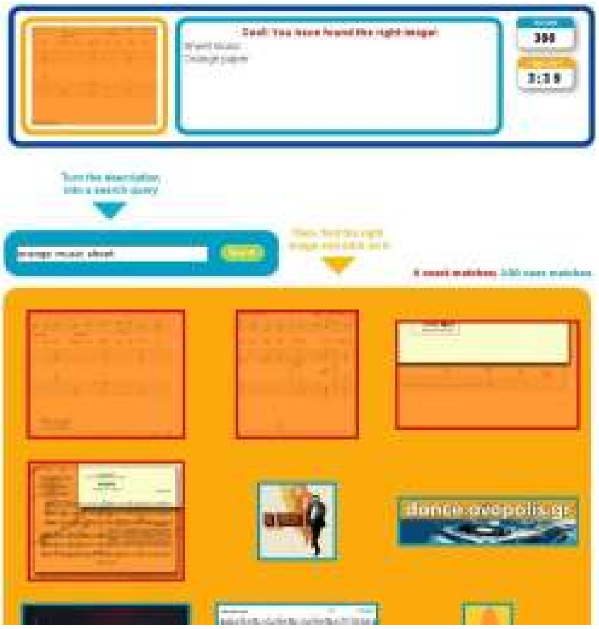
A major challenge with information availability as a research problem is the need for a cost-effective procedure to evaluate candidate solutions. As Voorhees points out, even minor changes to the Cranfield abstraction in order to evaluate interactive information retrieval result take a severe toll on cost-effectiveness of evaluation [13]. User studies are expensive!

An alternative approach follows the "games with a purpose" agenda proposed by Von Ahn [14]. This approach uses games to motivate people to perform information-related tasks, and has been applied successfully to such tasks as image tagging and optical character recognition.

There is even a game that evokes the information availability problem. In Phetch [15], users assume one of two roles, *seekers* and *describers*. The seekers compete to find an image based on a text description provided by the describer. The describer's goal is to help the seekers succeed, while the seekers compete with one another to find the target image within a fixed time limit, using search engine that has indexed the images based on tags generated from yet another game. In order to discourage random guessing, the game penalizes seekers for wrong guesses. The figure above shows an example of a seeker's view of the game.

The Phetch game does not include the possibility of negative success—the target image is always available. But it would be straightforward to adapt the Phetch game so that the target image was removed from all result sets returned to a seeker. The game would need an additional feature—the option for a seeker to assert that the image is unavailable. Correctly making this assertion would lead to success; incorrectly making it would be penalized.

An appealing aspect of games with a purpose is that they wrap realistic tasks inside a highly adaptable framework that yields quantitative results. Such a framework may be particularly suitable for the information availability problem.



## 8. CONCLUSION

While most of the work on information retrieval has focused on ranked retrieval and precision in the top ranked results, the information availability problem offers a realistic but general scenario that emphasizes recall. We have proposed evaluation criteria and ideas for tools that seem promising for addressing it. Finally, we believe that the games with a purpose framework offers the possibility of cost-effective evaluation.

## 9. ACKNOWLEDGMENTS

The author thanks Endeca for supporting this work. Specifically, colleagues Joyce Wang and Vladimir Zelevinsky developed the prototype shown in Section 6.

## 10. REFERENCES

[1] Saracevic, T. 2007. Relevance: A review of the literature and a framework for thinking on the notion in information science. *JASIST 58(3)*: 1915-1933.

[2] Van Rijsbergen, C. J. 2004. *The geometry of information retrieval*. New York: Cambridge University Press.

[3] Mizzaro, S. 1997. Relevance: The Whole History. *JASIST 48(9)*: 810-832.

[4] Zobel, J., Moffat, A., and Park, L. 2009. Against Recall: Is It Persistence, Cardinality, Density, Coverage, or Totality?. *SIGIR Forum 43(1)*:3-15.

[5] Dostert, M. and Kelly, D. 2009. Users' stopping behaviors and estimates of recall. In *Proc. of SIGIR 2009*: 820-821.

[6] Chen, H. and Karger, D. 2006. Less is more: probabilistic models for retrieving fewer relevant documents. In *Proc. of SIGIR 2006*: 429-436.

[7] Voorhees, E. 2004. Measuring ineffectiveness. In *Proc. of SIGIR 2004*: 562-563.

[8] Kuhlthau, C. 1993. A principle of uncertainty for information seeking. *Jour. of Documentation 49(4)*: 39-55.

[9] Wilson, T. D., Ford, N., Ellis, D., Foster, A., and Spink, A. 2002. Information seeking and mediated searching. Part 2: uncertainty and its correlates. *JASIST 53(9)*: 704-715.

[10] Fuhr, N. 2008. A probability ranking principle for interactive information retrieval. *Information Retrieval* 11(3): 251-265.

[11] Furnas, G., Landauer, T., Gomez, L., and Dumais, S. 1987. The Vocabulary Problem in Human-System Communication. *CACM 30(11)*: 964-971.

[12] Ma, H., Chandrasekar, R., Quirk, C., and Gupta, A. 2009. Page hunt: improving search engines using human computation games. *Proc. of SIGIR 2009*: 746-747.

[13] Voorhees, E. 2006. Building Test Collections for Adaptive Information Retrieval: What to Abstract for What cost? In *Proc. of First International Workshop on Adaptive Information Retrieval (AIR)*.

[14] Von Ahn, L. 2006. Games with a Purpose. *IEEE Computer 39(6)*: 92-94.

[15] Von Ahn, L., Ginosar, S., Kedia, M., Liu, R., and Blum, M. 2006. Improving accessibility of the web with a computer game. In *Proc. of SIGCHI 2006*: 79-82.

# Exploratory Search Over Temporal Event Sequences: Novel Requirements, Operations, and a Process Model

**Taowei David Wang, Krist Wongsuphasawat, Catherine Plaisant, and Ben Shneiderman**
Department of Computer Science
University of Maryland, College Park, MD 20742
{tw7, kristw, plaisant, ben}@cs.umd.edu

## ABSTRACT

Developing a detailed requirement analysis facilitates the building of interactive visualization systems that support exploratory analysis of multiple temporal event sequences. We discuss our experiences with collaborators in several domains on how they have used our systems and present a process model for exploratory search as the generalization of our experiences. This process model is intended as an outline of high-level analysis activities, and we hope can be a useful model for future and on-going exploratory search tools.

## INTRODUCTION

Developing hypotheses about relationships among temporal events and assessing their plausibility are important exploratory tasks in a variety of domains. These tasks can be broken down roughly in two parts: (1) discovering notable event sequences, and (2) evaluating the prevalence of such sequences to strengthen analysts' confidence in their hypotheses.

To this end, several interactive visualization approaches have been proposed to support exploratory analysis in temporal event sequences: business intelligence and financial fraud detection [6], clinical care and medical research [1][3][4][10], and web session logs [2]. These approaches seek to solve the problems analysts face when using a command-line query interface or a pure data-mining approach. However, these approaches have significant differences in their support for interactive exploratory analysis. In particular, they have different support for aggregation, comparison, and advanced exploratory search features over temporal categorical data.

This paper focuses on analysis tasks, requirements, and designs for event sequences (e.g. database of electronic health records that contain diagnoses, treatments, interventions, and admission/discharge information, etc.) We introduce two prototype visualization systems: Lifelines2 [9][10] (Figure 1) and Similan [11] (Figure 2). Because the two systems are at different stages of development, and apply different strategies, they support different requirements. We discuss the requirements for exploratory analysis over this type of data, and how these systems address these requirements. We then discuss how our case study users utilize these strategies. Finally, we draw from our users' experiences to present a preliminary process model of information seeking in the context of event histories.

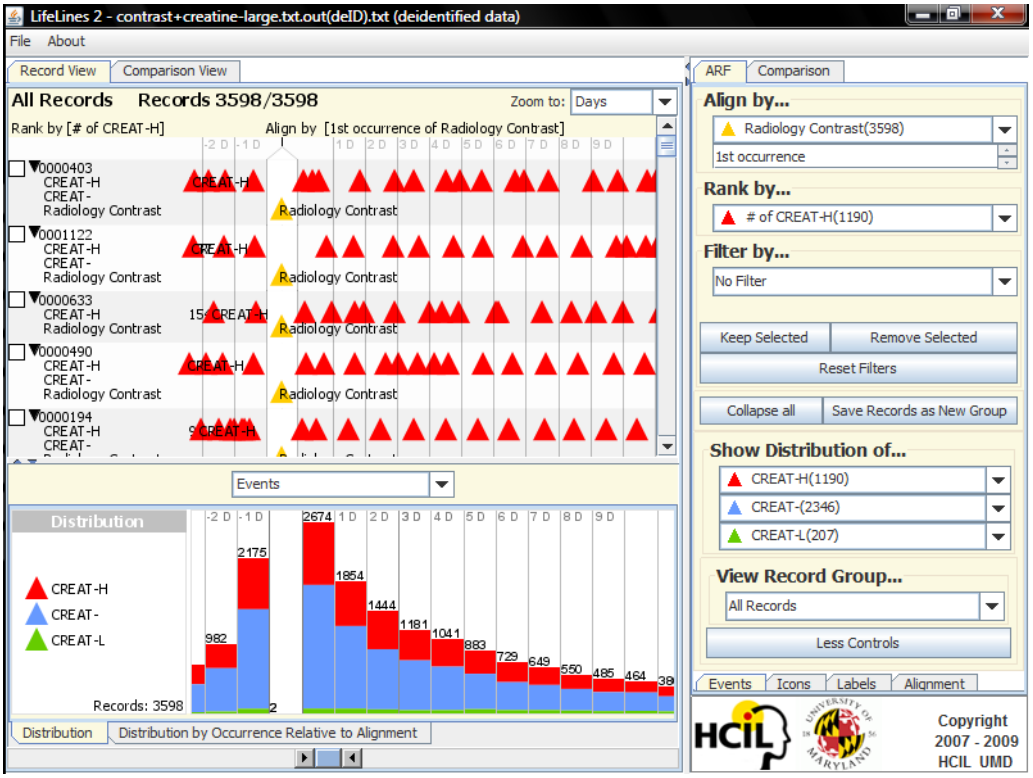
## SENTINEL EVENTS, ALIGN, RANK, AND FILTER

In many situations, domain analysts have a question regarding a particular event. We call this central event "sentinel event". Analysts may seek (1) what are the most commonly occurring events immediately prior to or after the sentinel event, (2) what is the distribution of another event with respect to the sentinel event, (3) or study the length of time between a sentinel event and another event. For example, clinical researchers may be interested in the distribution of mammogram procedures in all patients, prior to their diagnosis of breast cancer, and also seek the average length of time between first diagnosis of cancer and the time of death is.

However, visualizations typically do not provide analysts a way to rearrange the data around sentinel events for a more effective presentation. Instead, the data is often fixed on a linear time line, making sentinel events, which can occur anywhere, hard to spot.

To address this problem, we designed the *alignment* operator. Alignment allows analysts to dynamically re-center the data around a sentinel event across all event histories. This allows patterns specific to the sentinel event stand out. In Figure 1, all histories are centered on the sentinel event *1st Radiology Contrast* (yellow triangles), obviates all events around the sentinel event. When histories are aligned, the calendar is set to be relative to the alignment instead of on absolute dates.

In Lifelines2 and Similan, analysts can specify a sentinel event by choosing the n$^{th}$ first or last event of a certain type. Additionally, they can also specify all events of a certain type to be all be sentinel events. This multiple alignment allows analysts to study distribution of events near to all occurrences of a specific type.



LifeLines 2 - contrast+creatine-large.txt.out(deID).txt (deidentified data)

All Records    Records 3598/3598    Zoom to: Days

Rank by [# of CREAT-H]    Align by [1st occurrence of Radiology Contrast]

Align by... Radiology Contrast(3598) · 1st occurrence

Rank by... # of CREAT-H(1190)

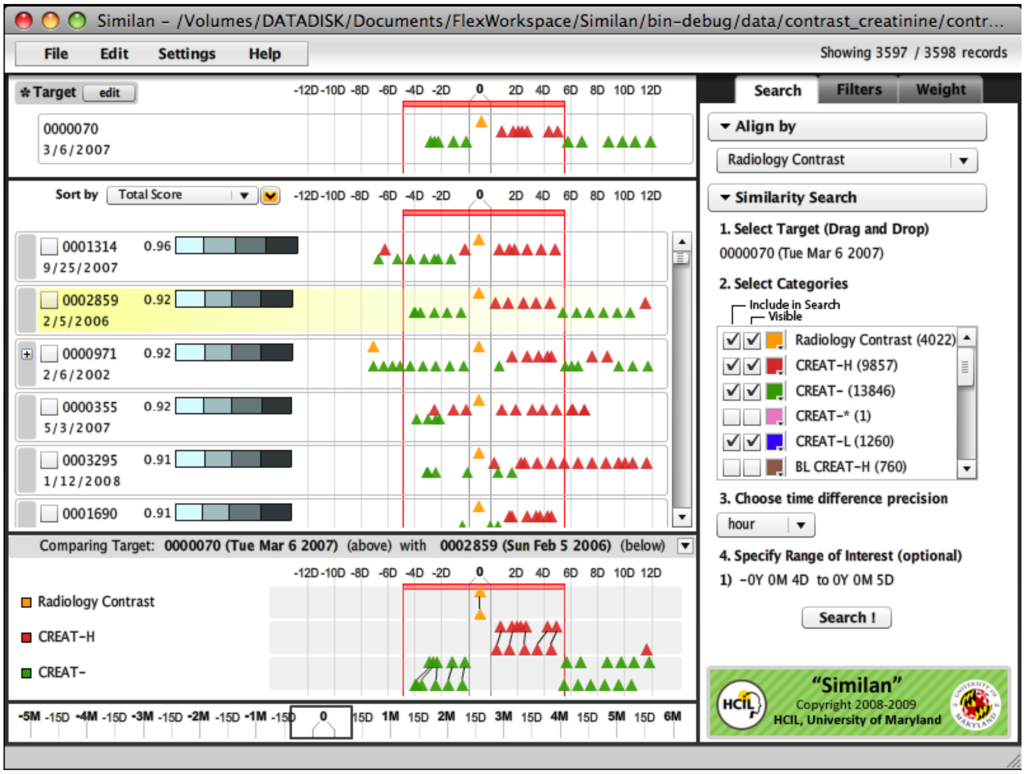
Filter by... No Filter

Show Distribution of... CREAT-H(1190) · CREAT-(2346) · CREAT-L(207)

View Record Group... All Records

Records: 3598

**Figure 1.** Screen shot of the Lifelines2 interface. The right portion is the control panel for a variety of operators. Top left is the main visualization panel, where each event history is shown as a horizontal strip on a time line. Each individual event is shown as a color-coded triangle (one event type is one color). The view shows that all histories are aligned by "Radiology Contrast" (the yellow triangles). The bottom half shows the temporal summary view of the red, blue, and green events over the visible time frame.



Similan – /Volumes/DATADISK/Documents/FlexWorkspace/Similan/bin-debug/data/contrast_creatinine/contr...

Showing 3597 / 3598 records

Target: 0000070 3/6/2007

Sort by Total Score

| Record | Date | Score |
|---|---|---|
| 0001314 | 9/25/2007 | 0.96 |
| 0002859 | 2/5/2006 | 0.92 |
| 0000971 | 2/6/2002 | 0.92 |
| 0000355 | 5/3/2007 | 0.92 |
| 0003295 | 1/12/2008 | 0.91 |
| 0001690 | | 0.91 |

Comparing Target: 0000070 (Tue Mar 6 2007) (above) with 0002859 (Sun Feb 5 2006) (below)

Align by: Radiology Contrast

1. Select Target (Drag and Drop)
0000070 (Tue Mar 6 2007)

2. Select Categories

| Include in Search | Visible | Category |
|---|---|---|
| ☑ | ☑ | Radiology Contrast (4022) |
| ☑ | ☑ | CREAT-H (9857) |
| ☑ | ☑ | CREAT- (13846) |
| □ | □ | CREAT-* (1) |
| ☑ | ☑ | CREAT-L (1260) |
| □ | □ | BL CREAT-H (760) |

3. Choose time difference precision
hour

4. Specify Range of Interest (optional)
1) -0Y 0M 4D to 0Y 0M 5D

File
Edit
Settings
Help
Target
edit
Search
Filters
Weight
Similarity Search
-12D -10D -8D -6D -4D -2D 0 2D 4D 6D 8D 10D 12D
Radiology Contrast
CREAT-H
CREAT-
-5M -15D -4M -15D -3M -15D -2M -15D -1M -15D 0 15D 1M 15D 2M 15D 3M 15D 4M 15D 5M 15D 6M
Search !
"Similan"
Copyright 2008-2009
HCIL, University of Maryland

**Figure 2.** Screen shot of Similan. The right portion is the control panel. The left portion contains three major panels. The center panel is the visualization of all event histories. The top panel shows the target history the user has selected. All histories in the center panel are ranked by their similarity to the target. The similarity scores are represented by color-coded bars. The bottom panel shows the comparison between the target against a currently selected history (shown in yellow background in the center panel). The user has selected a timeframe (red rectangular region) over which the match algorithm operates.

In Lifelines2, the alignment operator is complemented with more traditional information visualization operators: rank and filter. Analysts can rank all event histories by, for example, the number of occurrences of high-blood pressure diagnoses, reordering the most severe patients to be on the top of the list.

There are two modes for filter. Analysts can filter in the similar manner as rank by specifying a number of occurrences of a specified event type. All histories that do not have at least that number of that event type will be filtered out. Analysts can optionally designate the occurrences of these events to be only before or after a sentinel event. Secondly, analysts can specify a pattern of events to filter out histories that do not contain such pattern in an efficient manner [8]. An event pattern is a temporally ordered sequence of events or absence of events that analysts are interested. For example, analysts can use filter to find all patients who "were diagnosed with high-blood pressure, followed by no diagnosis of heart attack before a stroke."

## FINDING SIMILAR TEMPORAL EVENT SEQUENCES

The align, rank, and filter are the basic operators that allow analysts to study events of high interest and to find related events. However, sometimes analysts are interested in finding temporal event sequences that are similar to a specific history. For example, when a physician encounters a patient with symptoms that are rare and treatment options unknown, the physician may want to find past patients who share similar symptoms or medical history, and investigate the outcomes of different treatments.

This specific type of search has two main components. Analysts must specify what portion of a history is important, and what similarity means. In Similan, analysts would first picks a target history, and then choose a range on the time line to select a portion of that history that is relevant. The similarity matching is broken down to two parts. Similan first uses the Hungarian Algorithm [11] see how each history best matches the target. After the matches are found, Similan then assigns a similarity measure based on the number of mismatches and the "cost" of the match (based on temporal distance). Analysts can adjust the importance of mismatches. Analysts can also adjust the importance of out-of-order matches or matches with a large temporal differential.

Every history is then assigned a similarity measure, and displayed in descending order so that the most similar ones are on the top of the list. This is similar in spirit to the Rank-by-Feature framework, and allows analysts to review all histories before fine-tuning their search criteria. Analysts can review a similarity search and adjust the parameters of the similarity measure as described above to better suit their purposes.

The similarity search is further augmented to support "custom records". This means that analysts can manually specify a pattern to search instead of having to find one from an existing history.

## GROUPING, SUMMARY, AND COMPARISON

A natural extension to the variety of search mechanisms is to form subsets of histories for comparison. For example, hospital administrators may compare the differences of red blood cell counts for emergency room patients who experienced trauma and those who had not.

In Lifelines2, result of any filter operation can be explicitly made into a group. Analysts can choose to view any existing groups. They can also aggregate events for each group by using temporal summaries. Temporal summaries are stacked bar charts, where each stack represents one event type, aggregated over all histories. Analysts can examine the distribution of multiple event types at a glance [9]. The summaries are naturally integrated with alignment, so analysts can examine aggregations with respect to sentinel events.

Using temporal summaries, analysts can perform comparison among groups. A typical usage is to create two mutually exclusive groups and then put them side-by-side to study the temporal trend differences. A second use case is to successively narrow down a group of event histories and create successively smaller groups. Examining these groups' summaries gives analysts insight on whether this exploratory search path is on the right track.

## THE EXPLORATORY PROCESS MODEL

From working with our collaborators in medicine, student academic records, and law enforcement on drug trafficking phone records, we offer a preliminary process model of how our collaborators use our information visualization systems. Although the preliminary process model has numbered steps, our collaborators typically traverse in steps 2-4 in a pattern that is often not sequential.

1. Examine data for confidence (overview)
2. Exploratory Search
   a. Iteratively apply visual operators
   b. Evaluate results of manipulation
   c. Deal with unexpected discoveries
3. Analysis, Explanation
   a. Examine paths of search as a whole
   b. Determine to what extend are the questions answered
      i. At the limitation of the system
      ii. At the limitation of the data
   c. Refine existing questions
4. Report results to colleagues
   a. Document findings
   b. Disseminate subsets of data
5. Move onto new questions

One of the most common results of users looking at their own data through a visualization technique for the first time is the surprise that there are artifacts in the data (systematic errors, lack of consistency, etc.). This is because they have never seen it in an effective format before. As such, our collaborators would cursorily browse the data to make sure the data reflects what they know.

After gaining confidence of the visualization and of the data, they would start seeking answers to their pre-conceived questions. However, new questions often spawn when they notice interesting or unexpected data. At this point they would utilize their domain knowledge to try to explain what they see, or they would write down the new question for later exploration. We noticed that analysts may apply alignment on different sentinel events in the same exploratory session to look at the data in different views. They would actively manipulate the display by ranking, filtering iteratively, or change how similarity is weighted in Similan's search. However, alignment remains the strongest indicator on what focus they have on the data.

We found that aggregation techniques such as temporal summaries allow the analysts to look at the data quickly. Many of them learned to visually focus only on the summaries. They would also inspect the previously created groups by comparing their summaries to see qualitatively what kind of progress they have been making, and decide whether the path they are taking has potential. When they see a view of the data that answers their questions or contain interesting discoveries, they would save the state of their progress – saving the current group, and taking screen shots.

Although the process model we present here is still very preliminary, it already suggests elements that are indispensible to exploratory search in temporal categorical data. The first is a way to "anchor" the visualization for a particular path of search (like *alignment*), and allow analysts to quickly and dynamically change the anchor. The second is an overview of the entire dataset so that a mental model can be built quickly as the data is being manipulated. Next, a way to explicitly track users' steps of exploration is important. Finally, features that support viewing and comparison of different steps of exploration are critical to backtracking and taking excursions in the search process. We recommend these features for future applications.

## DISCUSSION

Performing exploratory analyses using a command-line query tool suffer from the problem that users have no mental model of the data. As a result, users have a hard time making judgments on how to refine their exploratory steps. Similarly, in a pure data-mining approach, lack of a mental model of the data makes interpretation of the results tricky. Information visualization allows opportunities for users to orient themselves at each step of the exploratory search, and enables maintenance of a consistent mental model throughout the process.

This paper presents several visualization and interaction techniques to let users control their exploratory paths and sustain a working mental model in searching temporal events. We argue that these approaches are more amenable to exploratory search. Information retrieval applications on temporal data can leverage work presented here to provide users a more fulfilling search experience. We discuss a preliminary process model for event sequences, and we hope to see interactive visualization techniques to be used in conjunction with information retrieval or data mining techniques to connect to their users as in [5][6].

## ACKNOWLEGEMENT

This project is supported in part by the Washington Hospital Center and Harvard - Partners HealthCare.

## REFERENCES

1. Fails, J. Karlson, A., Shahamat, L., and Shneiderman, B., A visual interface for multivariate temporal data: finding patterns of events over time", *Proc.IEEE VAST*, 2006
2. Lam, H., Russell, D. M., Tang, D.,Munzner, T., Session viewer: supporting visual exploratory analysis of web session logs. *Proc. IEEE VAST*, 2007
3. Plaisant, C., Lam, S., Shneiderman, B., Smith, M., Roseman, D., Marchand, G., Gillam, M., Feied, C., Handler, J., and Rappaport, H., Searching electronic health records for temporal patterns in patient histories: a case study with microsoft amalga," *Proc. AMIA Annual Fall Symposium*, 2008.
4. Ong, J., DataMontage Software, http://www.stottlerhenke.com/datamontage, 2006.
5. Post, A. R., Harrison, J. H.. Protempa: A method for specifying and identifying temporal sequences in retrospective data for patient selection. *JAMIA*, 2007.
6. Shahar, Y., Cheng, C., Intelligent visualization and exploration of time-oriented clinical data. *Proc. HICSS 1999*.
7. Suntinger, M., Schiefer J., Obweger H., and Groller, M.E. The event tunnel: interactive visualization of complex event streams for business process pattern analysis. *Pros. of IEEE Pacific Visualization Symposium '08*, 111-118, 2008.
8. Wang, T.D., Deshpande, A., Shneiderman, B., A temporal pattern search algorithm for personal histories, Tech Report # HCIL-2009-14, 2009, http://hcil.cs.umd.edu/trs/2009-14/2009-14.pdf.
9. Wang, T.D., Plaisant, C, Shneiderman, B., Spring, N., Roseman, D., Marchand G., Mukherjee, V., and Smith, M., Temporal summaries: supporting temporal categorical searching, aggregation and comparison. To appear in *Proc. IEEE Infovis, 2009*.
10. Wang, T.D., Plaisant C, Quinn, A., Stanchak, R., Shneiderman B., and Murphy, S., Aligning temporal data by sentinel events: discovering patterns in electronic health records. *Proc. CHI, 2008*.
11. Wongsuphasawat K. and Shneiderman B., finding comparable temporal categorical records: a similarity measure with an interactive visualization. To appear in *Proc. IEEE VAST, 2009*.

# Keyword Search: Quite Exploratory Actually

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

## ABSTRACT

This short position paper describes some evidence found that counters the argument that there are better ways to support exploratory search than keyword search. Instead, this paper suggests that keyword search actually provides people with the freedom to search in relation to their own current state of understanding, rather than in the terms controlled by a search system. The challenge for future exploratory search systems, therefore, may be to maintain and enhance such freedoms.

## INTRODUCTION

Some of the main arguments for research into exploratory search are that there are times when keyword search is not sufficient to support users. Such occasions include times when users who are unsure about a certain domain of information, uncertain about the terminology used by a search system, or unsure, even, about their own information needs [7]. Alternatively, therefore, many have been trying to support users in more exploratory conditions with alternative visualizations and user interfaces. Faceted browsing, clusters, and tag clouds, for example, are techniques that are designed to expose the structure of, or relationships within information to users, so that they can better understand a domain of information.

So why is it that keyword search persists? In some occasions, as described below, users have even preferred keyword search during exploratory tasks. While this may be because of people's familiarity with keyword search, the argument being made here is that exploration involves activities for which keyword search can be quite appropriate. The core of these learning activities, for example, is in making sense of how unfamiliar information fits in with a user's current understanding. It is potentially important therefore, that exploration allows users to freely express their current understanding. Further, however, hypothesis testing is also an important aspect of sensemaking, where searchers, as they learn, may want to see how results change according to their own ideas and developing conclusions.

## EVIDENCE FOR KEYWORD SEARCH

Above, it is suggested that there is some evidence for when users have preferred keyword search for more exploratory activities. In our own research, for example, we have seen that users found the facets in the mSpace browser useful more often for expressing multiple compound constraints in queries, than during exploration [10]. In another study, Capra et al. compared the RB++ browser and an un-configured Endeca[1] interface to the Bureau of Labor Statistics website[2]. First, the website, which of course has been designed for the dataset, performed well for all tasks. Further, however users specifically noted, during exploratory tasks, the lack of keyword search in the RB++ browser [3] (now included in the latest version).

More recently, our own research has created an analytical evaluation method [11] that can inspect search interface designs for how they support users in each of 16 searcher conditions. This method was used to evaluate, for example, the interfaces in the above examples [8]. Further, Google's keyword search was analysed, as shown in Figure 1, where the 16 profiles are described in Figure 2. These 16 search conditions range from users who know exactly what they want, and how to describe it (profile 16) to those who are learning and do not know what they will find (profile 1) [1].

In Figure 1, it might be noticed that the least supported searcher profile by keyword search is not profile 1, but profile 5, where users are scanning for an unknown document to take away, by recognizing it when they see it. This represents more browsing behaviour, where the user is trying to use keywords to describe a particular target that they are hoping exists. The support for exploration, however (towards profile 16) actually increases. Conversely, the most supported profiles are those where the user is trying to find a known target, by recognizing, and using keywords in their head. This process is actually better supported, with the help of query suggestions and spelling corrections, than users who know exactly what they want and can specify it, where users have to pick the terms that will most likely put their desired target at the top of the results list.

## SENSEMAKING

Making sense of information revolves around a user bridging a gap between their own knowledge and new information they have found [4]. In analysing how people hand-off information from one person to another, during

[1] http://www.endeca.com

[2] http://www.bls.gov

shift changes for example, pitching information at the right level of knowledge and understanding for the receiver, is important [6]. During any sensemaking process, therefore, it should be important to see how their own state of knowledge, however superficial, affects results.

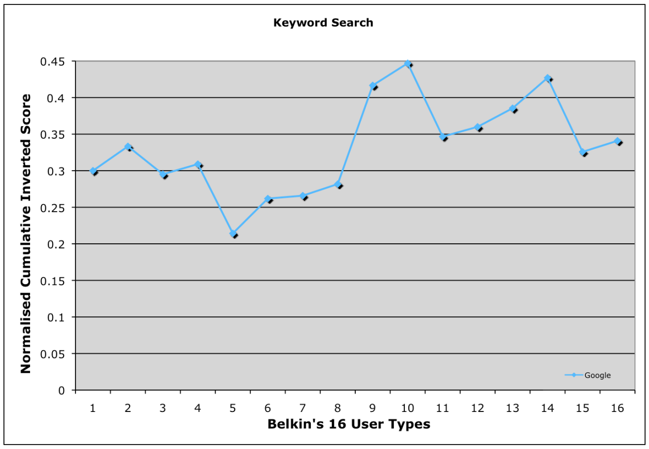
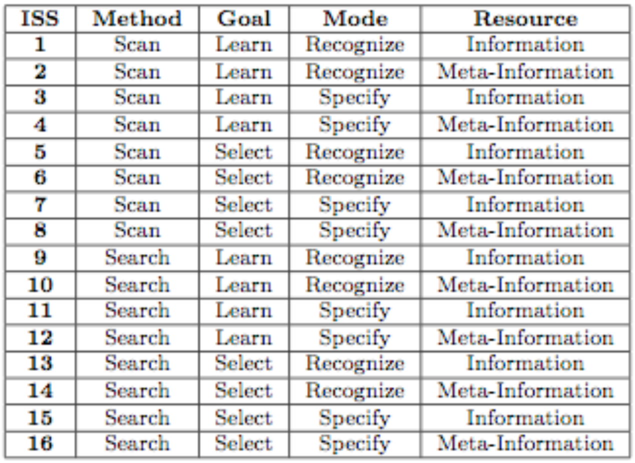


**Figure 1: An analysis of keyword search across different searcher profiles, where 16 is the most knowledgeable about their target, and profile 1 represents those learning and exploring [8].**



| ISS | Method | Goal | Mode | Resource |
|---|---|---|---|---|
| 1 | Scan | Learn | Recognize | Information |
| 2 | Scan | Learn | Recognize | Meta-Information |
| 3 | Scan | Learn | Specify | Information |
| 4 | Scan | Learn | Specify | Meta-Information |
| 5 | Scan | Select | Recognize | Information |
| 6 | Scan | Select | Recognize | Meta-Information |
| 7 | Scan | Select | Specify | Information |
| 8 | Scan | Select | Specify | Meta-Information |
| 9 | Search | Learn | Recognize | Information |
| 10 | Search | Learn | Recognize | Meta-Information |
| 11 | Search | Learn | Specify | Information |
| 12 | Search | Learn | Specify | Meta-Information |
| 13 | Search | Select | Recognize | Information |
| 14 | Search | Select | Recognize | Meta-Information |
| 15 | Search | Select | Specify | Information |
| 16 | Search | Select | Specify | Meta-Information |

**Figure 2: The 16 searcher profiles from Belkin et al. [1].**

People's own terms can also have a significant affect on memory and information processing. In a study of recalling blogs that participants had previously tagged, Budiu noted that participants performed best when they had tagged it using their own terms rather than the terms within the blog itself [2]. One possible hypothesis from these results is that users may perhaps struggle to interact with unfamiliar terminology laid out in faceted classifications, when they might rather try to communicate their own state of understanding. Even within facets of metadata, users are given the task of trying to find metadata they recognise, which, for all they know, may not be a valid option within the facet. At this point, it may be less effort for the user to say 'this is what I know', which is undoubtedly the way conversations would go when seeking the support of experts, or librarians as it used to be. There may be, in fact, no simpler way to express one's knowledge than to enter terms they understand into an empty box.

## SO WHAT DOES THAT MEAN FOR HCIR?

The root of the argument being built here, is that free-text search, is so called because it gives people freedom. The challenge for exploratory search and HCIR, therefore, is to try and maintain or incorporate *freedom* into interface designs or new visualizations. With many HCIR interface features, like faceted browsing, involving classification schemes built from the data or constructed from the domain of information, this may be challenging. Clustering engines, for another example, cluster around the data or metadata, and cluster labels could mean nothing to the user at all. While it is not uncommon for facets to be filtered by keyword searches [5], or, as in mSpace, for highlights to appear in facets, which relate to a result found in a keyword search [9], it might be of more exploratory value to provide stemming and support for synonyms to highlight related terminology in facets.

Another challenge for HCIR design, based on what we know of sensemaking and handoffs, maybe to monitor users and then try to pitch information at their level. It might be that dynamic faceted systems, which select the appropriate facets to show at any one time rather than simply all possible facets, may meet this requirement to some extent already. It might also be possible, however, to modify the terminology in facets, or vary the language in result lists, to terms that the user would understand. Understanding users though, of course, is a hard challenge.

I by no means have the answers here, but the core of the challenge to the HCIR community will be to properly, beyond the hypothesis of a position paper, investigate the question: why is it that keyword search persists, and is often helpful for exploratory search? It will be this discovery that will allow us to try and replicate the benefits in future designs. Until then, however, the challenge is, while leveraging the benefits of metadata, to try making freedom the core of our human computer interaction designs for information retrieval.

## CONCLUSIONS

The aim of this position paper has not been to suggest that the study of exploratory search is not important, or that research into alternative visualizations is not important. There are times, for example, especially where multiple or explicit constraints might be applied, such as in e-commerce, where faceted metadata is particularly useful. Instead, the aim of this position paper has been to highlight that there are elements of the keyword-response paradigm that are actually quite appropriate for exploratory search. While the challenge is to properly find out why keyword search has performed well in exploratory search, until then, the position here is that we should try to replicate keyword search's freedom in our future exploratory search designs.

## REFERENCES

1. Belkin, N.J., Marchetti, P.G. and Cool, C., Braque: design of an interface to support user interaction in information retrieval. Information Processing and Management, *29*, 3 (1993). 325-344.
2. Budiu, R., Pirolli, P. and Hong, L., Remembrance of things tagged: how tagging effort affects tag production and human memory. In *CHI'09*, ACM New York, NY, USA (2009), 615-624.
3. Capra, R., Marchionini, G., Oh, J.S., Stutzman, F. and Zhang, Y., Effects of structure and interaction style on distinct search tasks. In *Proc. JCDL 2007*, ACM Press (2007), 442-451.
4. Dervin, B., Foreman-Wernet, L. and Lauterbach, E. Sense-Making Methology Reader: Selected Writings of Brenda Dervin. Hampton Press, 2003.
5. Hearst, M., Design Recommendations for Hierarchical Faceted Search Interfaces. In *Design Recommendations for Hierarchical Faceted Search Interfaces* (2006).
6. Sharma, N., Sensemaking Handoff: When and How? Proceedings of the American Society for Information Science and Technology, *45*, 1 (2009). 1-12.
7. White, R.W., Kules, B., Drucker, S.M. and schraefel, m.c., Introduction. Communications of the ACM, *49*, 4 (2006). 36-39.
8. Wilson, M.L. An Analytical Inspection Framework for Evaluating the Search Tactics and User Profiles Supported by Information Seeking Interfaces, University of Southampton, 2009, 249.
9. Wilson, M.L., André, P. and schraefel, m.c., Backward Highlighting: Enhancing Faceted Search. In *UIST'08*, ACM Press (2008), 235-238.
10. Wilson, M.L. and schraefel, m.c., A longitudinal study of exploratory and keyword search. In *A longitudinal study of exploratory and keyword search*, ACM Press (2008), 52-56.
11. Wilson, M.L., schraefel, m.c. and White, R.W., Evaluating Advanced Search Interfaces using Established Information-Seeking Models. Journal of the American Society for Information Science and Technology, *60*, 7 (2009). 1407-1422.

# Using Twitter to Assess Information Needs: Early Results

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

## ABSTRACT

Information needs tell us why search terms are used, helping to disambiguate, for example, what exactly people are looking for with queries such as ‘Orange’ or ‘Java’. It is hard to understand goals and motivations, however, from the keywords entered into search engines alone. This paper discusses the pilot analysis of 180,000 tweets, containing search-related terms, to try and understand how people describe their own needs and goals. The early analysis shows that some terms academically associated with searching behaviours were infrequently used by twitter users, and that the use of terminology varied depending on the subject of search. The results also show that specific topics of searching tasks can be identified directly within tweets. Future analysis of the still on-going 5-month study will constitute more formal text analytical methods and try to build a corpus of real search tasks.

## INTRODUCTION

Search is a very loaded term. We seek, search, look, find, and explore information. Traditionally information retrieval has focused on matching keywords to documents, which we now see in most web search engines. Information needs, however, tell us whether searchers have entered ‘orange’ in order to find information about citruses, colours, or corporations. Further, information needs are typically part of larger work tasks [2], where the goal of searching for ‘orange’ may be to write a report, plan a food shop, manage a diet, or buy a phone, etc. Understanding information needs and work tasks, therefore, tells us whether interfaces need to be supporting activities such as: exploration, synthesis of information, comparison, or evaluation [11]. Further, understanding information needs tell us how we should design interfaces that support effective human computer interaction during information retrieval.

In this paper, the early stages of an analysis into how people describe and converse about their own information needs are presented. After discussing related work on information needs and analysing twitter, the method and results of this pilot stage analysis are presented. The paper concludes with some potential findings, before discussing the future plans for the full analysis of a 5-month archive of tweets.

## RELATED WORK

### Information Needs

Gaining insight into real information needs is not trivial. Advances however, have been made by, for example, studying search engine logs [4] and comparing keywords with relevance judgements [13]. Broder [1] noted that web searches typically fall under three categories of: transactional, navigational, and informational. Transactional queries are for web-based activities, such as buying, downloading, printing, etc. Navigational queries are simply to find a known website. Finally, Informational queries are those performed while trying to learn. Rose and Levinson [13] extended these into a hierarchy of goal types, such as types of learning, and types of transactions. Other research (e.g. [10]) has been trying to automatically infer goals based on click behaviour of a searcher over time.

The value of understanding information needs and goals is further emphasized by the inclusion of context when setting search related tasks in studies. TREC tasks [3], which are used to benchmark the performance of search systems, are created in association with *topics* so that it is clear what constitutes accurate results. Capra and Kules [9] further identified the types of contextual information that are important to provide to study participants when creating exploratory search tasks for user studies.

Jarvelin and Ingwersen, in discussing many aspects of information seeking, also noted that separate research areas have focused on both information needs and *perceived* information needs, where search is more closely related to how users currently understand their information needs [5]. Part of exploratory search and learning often involves first understanding a problem space, and then resolving it.

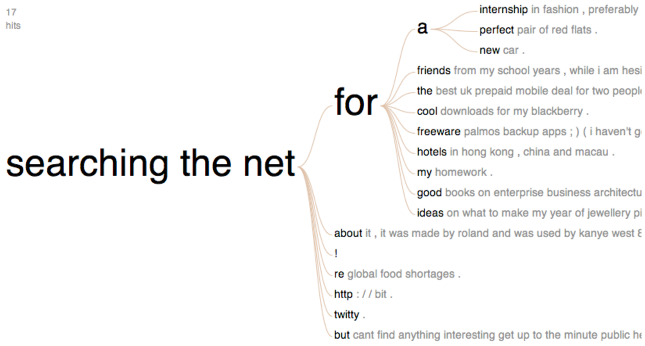

17
hits
searching the net
for
a
internship in fashion , preferably
perfect pair of red flats .
new car .
friends from my school years , while i am hesi
the best uk prepaid mobile deal for two peopl
cool downloads for my blackberry .
freeware palmos backup apps ; ) ( i haven't g
hotels in hong kong , china and macau .
my homework .
good books on enterprise business architectu
ideas on what to make my year of jewellery pi
about it , it was made by roland and was used by kanye west
!
re global food shortages .
http : / / bit .
twitty .
but cant find anything interesting get up to the minute public h

**Figure 1: Tweets that included the exact text: 'searching the net...', shown in a word tree.**

### Using Twitter as a Resource

Twitter is becoming a popular medium for communication, and recent work has begun analysing: networks, how people communicate, and what they talk about [6]. Pear Analytics, for example, classified tweets as being either: News, Spam, Self-Promotion, Pointless Babble, Conversational, or Pass-Along. Their results showed that around 40% was babble, 37% was conversational, and, in third place, Pass-Along constituted 9% of the tweets [7]. Similarly, the Web Ecology Project released a sentiment analysis of tweets regarding Michael Jackson's death [8]. In comparison to a typical archive of tweets, the Michael Jackson archive included a significantly larger portion of negative tweets.

## GATHERING INFORMATION NEEDS FROM TWITTER

With the aim of better understanding real information needs, Twitter was analysed as a worldwide resource of people's public discussions, to find conversation about searching behaviour. Although Pear Analytics said that twitter is mostly used for babble and conversation, these are the elements of their taxonomy, as opposed to news, spam, and self-promotion, that will provide value for this study. Figure 1 shows a basic example, where people used the exact words: 'searching the net…'. The analysis described here is of the first 2 weeks of a larger 5 month investigation into the ways people describe their own searches on twitter.

### Method

To gather tweets that describe searching behaviour, a Twitter search was automatically queried every hour for the most recent 100 tweets for each of the 10 search related terms[1] listed in Table 1. The terms, mainly selected from academic publications from search communities, were also passed through a thesaurus to identify and consider additional English language terms. Alternatives, as in those not used, were checked with a single search on twitter to assess current frequency of use on twitter. The chosen terms were those above a significant drop-off point. This process was performed for two weeks during this pilot analysis. To catch as wide a net as possible, all tweets including these terms were archived without any analysis of whether they were describing searches. That is, although Figure 1 shows a basic example of where people explicitly talk about searching the '[inter]net', this research has aimed to discover real-world information needs and work tasks, which may involve search behaviour in real or physical environments, as in Figure 3. Further, each of these terms were queried in their past, present, and future variations, such as the query 'find OR finding OR found'.

To analyse the tweets, several methods are being considered. The initial analysis here is designed to be more qualitative to a) reveal early interesting qualitative insights, such as in Figure 2, and b) help inform the way that the final dataset should be more formally analysed. Initially, for visualization, tag clouds were considered, however these revealed very little about what people searched for. After a more structured semantic analysis of tweets, however, tag clouds of identified *search topics* may provide interesting insights. At this stage, aside from some high-level statistics, Word Trees, using IBM's ManyEyes project [14], were used to manually and qualitatively explore the content.

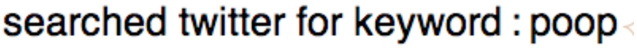

searched twitter for keyword : poop

**Figure 2: This exact phrase appeared in 2 separate tweets.**

### Results

In total, 189,452 unique tweets were captured from 163,564 authors. Additionally, 14,959 re-tweets were archived, where users echo the tweets of others to their own network.

**Table 1: Showing a breakdown of the tweets collected during the first 2 week archiving process.**

| Term | Unique Tweets | ReTweets | Authors |
|---|---:|---:|---:|
| Exploring | 21,287 | 1,414 | 19,119 |
| Finding | 26,333 | 1,107 | 25,656 |
| Foraging | 910 | 1,627 | 790 |
| Hunting | 26,534 | 1,123 | 22,666 |
| Investigating | 19,255 | 2,016 | 14,488 |
| Looking | 22,783 | 1,267 | 21,142 |
| Retrieving | 3,506 | 1,500 | 3,269 |
| Searching | 25,493 | 1,788 | 20,095 |
| Seeking | 15,767 | 1,380 | 12,987 |
| Studying | 27,584 | 1,737 | 23,352 |
| Totals | 189,452 | 14,959 | 163,564 |

#### *Frequency of term use*

One contribution of this analysis is to see the popularity of different terms as people describe their searching actions. 'Studying' was the most popular term used, but, despite being a popular metaphor for how people may search [12], the 'Foraging' term, and its temporal variations, were hardly used. Similarly, and perhaps surprisingly, the term 'Retrieving', and its variations, were used significantly fewer times than many of many of the other terms. The terms 'Searching', 'Hunting', and 'Finding' were also popular terms, but 'Hunting' in particular was often used in relation to sport, as discussed below. While 'Looking', as might be expected, was quite popular, two terms relating to exploration ('Exploring' and 'Investigating') were also quite popular. The term 'Seeking', while perhaps quite an academic term for search, was used almost half as often as most other terms, but significantly more than the term 'Retrieving'.

[1] Unfortunately, the term 'browse' failed during this pilot study, but has been fixed for the 5 month study.

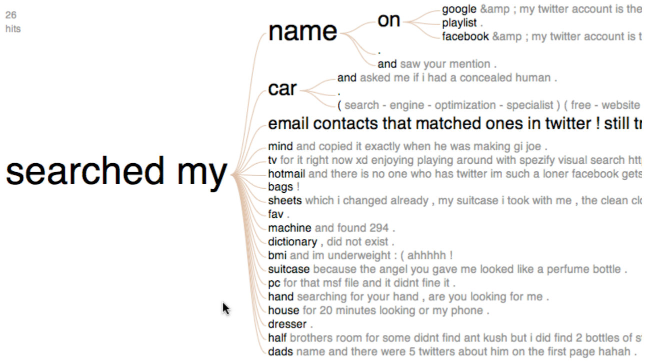

26
hits
searched my
name
on
google &amp ; my twitter account is the
playlist .
facebook &amp ; my twitter account is t
.
and saw your mention .
car
and asked me if i had a concealed human .
.
( search - engine - optimization - specialist ) ( free - website
email contacts that matched ones in twitter ! still tr
mind and copied it exactly when he was making gi joe .
tv for it right now xd enjoying playing around with spezify visual search htt
hotmail and there is no one who has twitter im such a loner facebook gets
bags !
sheets which i changed already , my suitcase i took with me , the clean cl
fav .
machine and found 294 .
dictionary , did not exist .
bmi and im underweight : ( ahhhhh !
suitcase because the angel you gave me looked like a perfume bottle .
pc for that msf file and it didnt fine it .
hand searching for your hand , are you looking for me .
house for 20 minutes looking or my phone .
dresser .
half brothers room for some didnt find ant kush but i did find 2 bottles of s
dads name and there were 5 twitters about him on the first page hahah .

**Figure 3: Tweets described searching behaviour in both physical and digital environments.**

*Language associated with terms*

Another contribution of performing this qualitative analysis, is in being able to see how different terms are used to describe different kinds of searching. Figure 4, for example, shows that the 'Finding' terms were often associated with finding an ideal or optimal results. When followed by the word 'my', however, the task was often re-finding, and usually for locating where technology was in the home. A third regular use of the 'Finding' terms was followed by the word 'out', which typically represented more exploratory tasks.

The variations of the term 'Exploring' were typically used in regards to new places, such as cities and neighbourhoods. Many people self-reported as exploring twitter for the first time. Exploration, however, was often associated with abstract objects, such as ideas, options, and possibilities, but also with genre's of music and film.

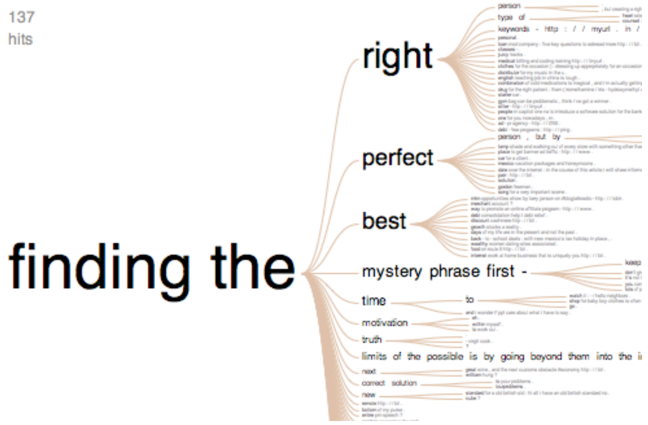


**Figure 4: Use of the term 'Finding' when followed by 'the'. This combination was often associated with ideal individual results, the 'right', 'perfect', or 'best'.**

Perhaps interestingly, the variations of 'Foraging' were nearly always used in conjunction with food terms. Although people rarely used the term, people self-reported as foraging in cupboards, fridges, kitchens, and freezers, with the aim of locating food at mealtimes. When not associated with food, the term foraging described behaviour in outdoor areas, such as yards or woods, but also within documents.

'Hunting', when not being used to discuss sport, was for: new jobs, people (including witches), and technology. Like the term 'finding', 'Hunting' was often used in association with adjectives representing optimal results, such as a best, cheapest, or perfect.

The 'Investigating' terms were typically used in relation to crimes. When used, however, they were often investigating the informational boundaries around such events, as investigating: claims, correctness, cause, and circumstance.

Like 'Hunting' and 'Finding', the term 'Looking' terms were often related to people, jobs, and technology, and their optimal variations, including 'best', 'right', and 'perfect'. The term was also used, however, in association with looking for a new place to live, excuses, and the original copies of objects. People also often described looking for entertainment items, such as music, books, and movies.

When used, 'Retrieving' terms were related to gathering lost or distant items, often one's daughter. The majority of subjects in these tweets, however, were digital, such as retrieving lost or archived passwords, records, files, pictures, and tweets.

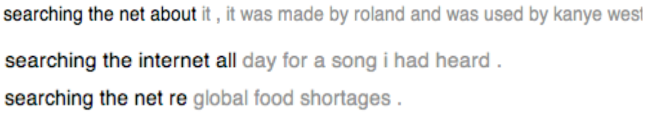
'Searching' terms were used for a large range of subjects. While sometimes used in relation to optimal (best, next, perfect) technologies, 'Searching' was also used for food, missing people, soul mates, truth, music, friends, and pictures. The 'Searching' terms, however, produced the highest number of exact quoted search terms, discussed below. The 'Searching' terms also returned the highest number of tweets that described venues for search, such as Google, Facebook, eBay, Twitter, etc.

When not used for adult advertisements, the term 'Seeking' was primarily used for finding people for jobs, or a place to stay. It was also heavily used with exploratory and abstract terms such as 'the truth' and 'to be understood'. 'Seeking' terms were also used in breaching peoples boarders, such as 'new lands' and 'faces'.

Finally, the studying term was primarily used when discussing forth-coming exams. Sometimes, however, studying was associated with self-driven learning on topics such as the bible, psychology, and photography. Consequently, the 'Studying' terms provide some interesting topics for learning tasks in studies, including the history of tobacco and the effects of erosion.

*Specific subjects of search*

Finally, a third contribution of the analysis is in identifying specific searching tasks. Figure 5, for example, shows three complicated self-reported information needs. The first represents a complex search need, where the user has two pieces of related information. The second and third represent more exploratory learning tasks. Figure 6, however, shows that many twitter users directly provided search terms they had used, using speech marks. Figure 6 indicates those that explicitly used the past-tense variation of ‘Searching’ followed by the word ‘for’ and then speech marks.

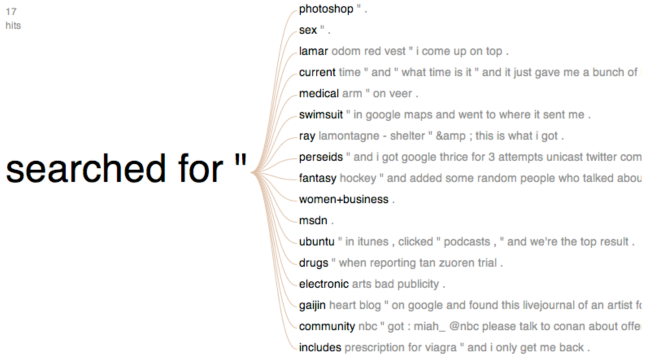


searching the net about it , it was made by roland and was used by kanye west

searching the internet all day for a song i had heard .

searching the net re global food shortages .

**Figure 5: One complex search task and two exploratory tasks described by twitter users.**



searched for "

photoshop " .

sex " .

lamar odom red vest " i come up on top .

current time " and " what time is it " and it just gave me a bunch of

medical arm " on veer .

swimsuit " in google maps and went to where it sent me .

ray lamontagne - shelter " &amp ; this is what i got .

perseids " and i got google thrice for 3 attempts unicast twitter com

fantasy hockey " and added some random people who talked abou

women+business .

msdn .

ubuntu " in itunes , clicked " podcasts , " and we're the top result .

drugs " when reporting tan zuoren trial .

electronic arts bad publicity .

gaijin heart blog " on google and found this livejournal of an artist f

community nbc " got : miah_ @nbc please talk to conan about offe

includes prescription for viagra " and i only get me back .

**Figure 6: Twitters often labelled, using speech marks, exact specific terms they had queried different services for.**

## CONCLUSIONS

This work has reported the early pilot analysis of a work-in-progress investigation into tweets that included searching-related terminology, archived in the first 2 weeks of a larger 5 month study. The analysis revealed early insights into how often, and in regard to which forms of search, different search terms were used by twitter users when discussing their own searching behaviours. Where previous research has typically tried to deduce information needs from search engine logs, this research is trying to identify information needs from publically available conversations on the web.

In completion of the full 5 month long study, more formal text-analysis techniques will be applied, perhaps including a sentiment analysis [8], to find out if, for example, the search behaviours that people feel are worth tweeting mainly surround difficult or novel searches. Further, such an analysis may be able to identify the frequency, subject, and success of different types of searching goals [13]. Part of the aim, therefore, will be in building a resource of realistic search tasks for different types of searching contexts, which can be used in future user studies, and informed by people’s own self-driven descriptions of searching behaviour. The research described here, however, provides early insights into how people describe and communicate their own searching activities to others. Understanding how people perceive their searching activities and needs can help inform the design of interfaces for human computer interaction during information retrieval.

## REFERENCES

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

[2] Byström, K. and Hansen, P. Work tasks as units for analysis in information seeking and retrieval studies. in Bruce, H., Fidel, R., Ingwersen, P. and Vakkari, P. eds. *Emerging Frameworks and Methods*, Libraries Unlimited, Greenwood Village, CO, 2002, 239-251.

[3] Harman, D.K. The TREC conferences. (1997). 247-256.

[4] Jansen, B.J. and Spink, A. How are we searching the World Wide Web?: A comparison of nine search engine transaction logs. 42, 1 (2006). 248-263.

[5] Järvelin, K. and Ingwersen, P. Information seeking research needs extension towards tasks and technology. 10, 1 (2004). 10-11.

[6] Java, A., Song, X., Finin, T. and Tseng, B., Why we twitter: understanding microblogging usage and communities. in KDD'07, (2007), ACM New York, NY, USA, 56-65.

[7] Kelly, R. Twitter Study - August 2009, Pear Analytics, 2009, 1-13.

[8] Kim, E., Gilbert, S., Edwards, M.J. and Graeff, E. Detecting Sadness in 140 Characters: Sentiment Analysis of Mourning Michael Jackson on Twitter, Web Ecology Project, Boston, MA, USA, 2009, 1-15.

[9] Kules, B. and Capra, R., Creating exploratory tasks for a faceted search interface. in HCIR'08, (2008).

[10] Lee, U., Liu, Z. and Cho, J., Automatic identification of user goals in web search. in WWW'05, (2005), ACM New York, NY, USA, 391-400.

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

[12] Pirolli, P. and Card, S., Information foraging in information access environments. in CHI'95, (1995), ACM Press, 51-58.

[13] Rose, D. and Levinson, D., Understanding user goals in web search. in WWW'04, (2004), ACM New York, NY, USA, 13-19.

[14] Viegas, F., et al. Manyeyes: a site for visualization at internet scale. IEEE Trans. Visualization and Computer Graphics, 13, 6 (2007). 1121-1128.

# Integrating user-generated content description to search interface design

Kyunghye Yoon

State University of New York at Oswego

Oswego, NY 13126

kyoon@cs.oswego.edu

## ABSTRACT

In this paper, the ideas discussed will focus on the integration of user tags into information search and interface design. There are two propositions: 1) user-created tagging is a valuable source of user's personal views and annotations that can augment the content description of information resources; 2) information search can be viewed as seeking meaning in information use and need. It is suggested to draw user meaning from the tag data by employing the topic and comment as two dimensions of linguistic meaning and to represent the meaning as a simple semantic relation that can be used for clustering the search results to supplement the traditional topic-based information matching. The sample analysis was done with the user tagging data positioned in the Delicious site to identify the semantic relations of linguistic meaning.

## 1. Introduction

Tagging is one of the fastest growing applications on the web and has been gaining popularity lately. Tagging allows users to label and assign terms to information objects for later access. The aggregated tags, a product of collaborative tagging, can not only be used as information organization and management tools for the user who created them, but also they can be shared by other users to search, browse and access information resources. The aggregated tags can add user meaning to the content to represent document content from the users' perspectives in addition to the original content. With the folksonmic advantage of free forming content description from the bottom-up, user tags are a vital source to help retrieval queries for the public and can augment the authoritative document organization and classification systems as well [4, 6]. This paper will discuss the incorporation of user tagging data into information search interface design for displaying search results to help supplement the traditional topic-based keyword searching.

Current search engines and information retrieval systems are based on the keyword matching of the terms presented between the document representation (i.e., surrogate) and the user query. Matching is done for each term independent of other terms in the user query or in the document content often causing the user meaning to be lost in the search process. With the constantly increasing growth of information resources on the Internet, information search on a topic usually results in a large number of information contents; this makes it necessary for the user to go through a long list of search results to find relevant ones or to refine the search. It has become a challenge to provide a meaningful user interface for users to effectively browse and filter out the search results.

One way to meaningfully present the content of the search result is with the use of semantic relations from user tagging data. Integrating user tags to the content description can help users browse and make relevance judgments. The interface will provide a search result that represents the users' descriptions of specific attributes and attached meaning from the user tags as well as providing additional topical identification. A simple semantic relation of user tagging is suggested by employing the concept of topic and comment to devise a meaningful user interface. Topic and comment is adapted from social linguistics as the two distinct components of linguistic meaning.

The following three sections present: 1) tagging as a user-created content description is useful to information representation, 2) information seeking and search can be seen as seeking meaning, and 3) constructing semantic relations of content representation from user tagging data is suggested with sample analysis of tagging data. A set of sample analyses will then follow.

## 2. User-generated content description for information search

Tagging is not merely a tool or mechanism for creating information structures but also a tool for reconceptualizing information architecture [3]. One of its values is in the process of freely describing and assigning labels to information resources, creating folksonomies. Through this process of categorizing and assigning terms, meaning can emerge from the users on the information content, which can add the personal specific user meaning attached to the content description beyond the generic (i.e., free of context) topic identification. The idea of a bottom-up process of a user's direct personal specific description can serve as an additional access mechanism for other users to share annotative reviews and narratives.

The aggregated tagging across the users opened up a new way for the public users to contribute to generations of content description; this information resource can augment the traditional information organization such as library classification systems produced by professionals and authorities [11]. The aggregated user content description is viewed as a valuable source to augment the traditional document representation of the rigid and unitary language model for digital libraries and second-generation web development [12]. According to the social constructionist view of information science, the traditional information description assumes that "documents have a substance [i.e., objectively identifiable meanings or messages that can be represented in a clear structure of terms (nouns)]" [12].

There have been a few attempts to make use of folksonomic characteristics of user tagging incorporated in the traditional and controlled vocabulary-based classification and representation schemes. Face tags is one example that shows a semantic approach to collaborative tagging by incorporating faceted classification schemes to facilitate multidimensional browsing where it is assumed that users provide the structure with a folksonomy [9]. Bubble up tags is another example in which aggregated terms of the most popular tags are assumed to represent the content [10]. Terms together in a group may indicate a semantic relationship and association among terms and can be a useful content description. Even though the co-occurrence of terms does not identify any explicit relationship, the value seems to be in the highly movable usage of the terms and their linguistic relations to a user group at specific points in time and space [2]. Overall, these studies suggested an implicit structure in the usage of terms in folksonomy and a rich source for metadata filters based on shared or divergent approaches to the categorization of knowledge [2].

Facets of tags and bubble up tags are an attempt to incorporate the multiple dimensions of words and their relations together, but they are limited in conveying linguistic meaning. They are carried within the topic-based information organization paradigm that assumes independent terms for information search rather than a meaning created by a set of terms with semantic relations.

## 3. Information seeking and meaning

Topic and comment, the two distinct linguistic components of a meaning, provide an approach to identifying a semantic relation of information seeking [13, 14, 15]. According to functional linguistics, meanings are complete by both topic and comment. Topic is what it is about and comment is what the speaker attaches and relates to the topic. The concept of topic and comment has been applied to information science and the traditional classification theory, which adopted topic as a dominant element to represent document content [1].

In the traditional information system with topic-oriented information organization, aboutness of a text is the core dimension for document representation and classification. This is seen in the current information organization and retrieval systems: the query of a user is a list of keywords, and document representation (i.e., surrogate) is composed of a list of terms that appear in the text (e.g., inverted index), both of which do not reflect the semantic relationships among the terms but the independent occurrences. The keyword matching has been criticized for the uni-dimensional and generic characterization of topic in the field of information science [7]. It is because topic alone is not a sufficient criterion and needs to be supplemented by other criteria such as situational factors.

The meaningful connection between a user need and the information content was sought in a study done of information seeking interaction that empirically examined topic and comment as necessary components of information need description [14, 15]. The study confirmed that both topic and comment are essential in user's information need articulation as the two orthogonal dimensions of information to meaningfully describe the user's information as well as to represent information content meaningful to a specific user need and use context. The sequence analysis of the user-source interaction showed that topic was employed first to set a common ground for the interaction and then comment provided aspects that stressed the specific use context such as the goal of the user's information seeking or the intended use of information [13, 14]. The term comment implies not only the discrete individual attributes of the use context but also the relations among them to the user meaning.

This suggests a strong basis for an argument that information search can be improved if the meaning of the information and the need are related by the two components of topic and comment together. Often users may not address both the topic and comment components when they search for information even though the connection of topic and comment is the full meaning represented in the content. It is because their cognitive state lacks the full meaning when they are in need (i.e., they do not know about something, thus they need to find out about it). But providing the topic and comment relations of the content of search result will be useful for the user's relevance check. Given that topic is first employed for the information need specification and then comment is the subsequent necessary component, it is suggested that the comment dimension should be considered as an addition to the content description of the search result, under the current topic-based matching paradigm.

## 4. Tag data in search interface design

The idea of utilizing the folksonomic description of user tagging data can be attained by representing the two linguistic components of topic and comment of the content as a simple semantic relation. It is a simple semantic relation with two nodes of topic and comment, and a connector, the relation between the two nodes. It is distinguished from the general semantic relation of organized terms to their corresponding hierarchical concepts, in an ontological sense which was applied in information research with explicit and logical associations of concepts and relations [8]. Most ontological relations were concerned with generic (i.e., free of context) and unitary topic-based relations, which were found of little utility in facilitating information retrieval [5]. Utilizing folksonomic descriptions seems to facilitate a stronger and dynamic engagement of user searching based on the use of contexts. It is suggested, in this paper, to take a simple relation of topic and comment from the user-generated content descriptions.

## 4.1 Analysis of tag data

The analysis was exploratory, initially investigating the possibility of inferring topic and comment relations from user tags created from content description. Even though words appearing together in a document may indicate a useful association, there is yet no explicit relationship identified "even if two tags were used in concert all the time by a wide variety of users across multiple resources, we couldn't make any claims about them other than that they are highly related" [10]. Therefore, the analysis was mainly exploratory investigation of capturing semantic meaning from the user by using semantic definitions and relations in natural language, which is expected to provide a basis for later automated inquiries. The analysis focused on the descriptive tags with notes of personal annotation and review among different kinds of tags such as resource type, source and ownership, descriptive and personal [10].

Delicious (del.icio.us) is chosen for a context of sample analysis because this site is one of the earlier social bookmarking sites, that has minimal limitations in the types of web sites users access and in the way they assign tags and note. A set of sample tagged websites (i.e., bookmarks) were intentionally chosen from articles in journals and blogs in order to retain the independent document unit value of information for content analysis. Table 1 in the appendix shows the four sample sites with top tags listed on the collected date. Overall, there were little explicit semantic relations among the top tags indicated from the list alone. Therefore the analysis was done at the individual user level with user tags and notes. User tags contain a list of single terms the user assigned to the bookmarked website. User notes are a full text of user meaning in natural language form that Delicious allows users to freely attach.

## 4.2 Example analysis

The analysis was focused on the co-occurrence of the tag terms within individual user's tag lists. User notes were particularly helpful to induce the user meaning associated with specific use of tag terms. Users used notes to remind themselves why they bookmarked the material, and why it was important. [1] Often, user notes included quotes from the original content or from the link where the article was located. It also included the user's own annotative descriptions and comments. In either case, it was in a natural language text, not a single word, to possibly present a semantic basis close to the user meaning. Thus it helped to understand a user's view and analysis of the user defined concept.

For each sample, a simple semantic relation was constructed from the content analysis of user notes and sets of tags. The intention was to create two different clusters of concepts for the topic and the comment. Even though comment dimension includes verb phrases, the cluster for comment was treated as a cluster of noun phrases similar to that of topic because most user tags were nouns. The relation inferred by verbs from user notes provided a connection between the two nodes. Figures 1 through 4 show the examples of the simple semantic relation of the tag terms. Each of the terms was selected from the top tags and placed in the circle while the cluster of related concepts was formed with connecting circles. Usually, the cluster of a concept included terms to represent the concept at multiple levels as the semantic progression in the text develops. The term is related to the broader concepts as it goes down. There are two main clusters: one of topic in the left and one of comment in the right.

Some of the terms were driven from user's specific attachment rather than presented in the original text. For example, in Figure 2, the concept of internet, hypertext, web and web2.0 was added by users even though the terms were not mentioned in the actual content: the article was forecasting such technologies. Arrows represent the relation between the clusters with the inferred relationship in quotation marks (usually a verb). It was also in line with the bubble up tags that some words tend to occur together. For example, "medicine" and "science" in example 3; and "internet" and "technology" in example 4; the two terms listed mostly together in each set of user tags.

The simple semantic relations resemble Hutchins' micro structure and macro structure of text semantic progression of the two components of theme and rheme [7]. Macro structure is a semantic relation representing the underlying propositions in the global semantic progression; whereas micro structure is a semantic association of specific and individual segments within the semantic coherence of the global progression [7]. Each cluster of the simple semantic relation from the analysis included terms in hierarchical progression, which is not necessarily the same as in ontological relation even though it does include a broader context of use. Some individual user tags were pertained to the micro structure (i.e., a part of the content rather than the whole text) but the count did not seem to be significant enough to reach the top tags across the user group. The broad level concepts are usually from user-created meanings related to the content such as application area and use dimension that were not included in the original content.

## 4.3 Suggestions

The simple semantic relation of topic and comment inferred from user tags can be applied to the user interface to provide document description with clustering to help users to better grasp the content in a search situation. Information search starts with one or a partial dimension of information (i.e., topic or comment) as an incomplete meaning with a few keywords because users do not know how to fully represent the need. The search is done by

[1] This came from user interviews for other related studies not published yet

matching the keyword(s) between the user query and the document surrogate. Then the search results are displayed and this can be done by consolidating both topic and comment dimension of the information content. It will be useful to display the document of the search result how the document content is related to the keywords used in the query in relation to its full meaning of the content.

The use of the semantic relation in grouping the search results is most effective. One specific way is to group the search results by this new dimension attached to the one used in search, which is found from the semantic relation. Taking Figure 1 as an example, the text can be matched to a query "simplicity" or "simplicity in UI design." In displaying the text as one of the search result, the description not only contains the "simplicity" how it matches the user query which is mostly done in the current search systems but also the other dimension, "overrated" or "doesn't sell" of the content. This new dimension will discriminate the text from others in the search results all of which will match the topic of "simplicity" but with a variety of diverse meanings such as "to improve design process" or "as critical design principle" to make up a few.

Another way is to use the terms from the user tags in matching and incorporate the semantic relations in displaying the results. For example with Figure 2, the text can be matched to a query, "Internet history," or "evolution of Internet" even though the text does not have the term Internet. Then this text will be grouped with those that foretell the Internet. Under the topic used in the search, "Internet history," the new dimension can be used to create clusters of search results other than those explicitly used in the user query.

## 5. Conclusion

The value of collaborated tagging was viewed in the creation of the aggregated user assigned information content description that can meaningfully connect other users to the information in the collection. The discussion addressed the nature of users' information behavior inherent in the meaning attached to information contents. An attempt was made to capture user-created meaning attached to the content from user tagging data with the implicit relations of their meaning that they were trying to connect with information objects. Topic and comment was used as the basis for simple semantic relations of the user tags. Sample analyses showed interesting evidence for further in-depth investigation of the user-created tagging.

## 6. Reference

[1] Begthol, C. (1986). Bibliographic classification theory and text linguistics: aboutness analysis, inter textuality and the cognitive act of classifying documents. *Journal of Documentation* 42(2), 84-113.

[2] Bruns, A. (2008). *Blogs, Wikipedia, Second Life and beyond: from production to produsage*. New York: Peter Lang.

[3] Campbell, G. D. & Fast, K. V., (2006) From pace-layering to resilience theory: The complex implications of tagging for information architecture, *Proceedings of IA Summit 2006* (Vancouver, March 23-27, 2006), ASIS&T. DOI=http://www.iasummit.org/2006/files/164_Presentation_Desc.pdf

[4] Golder, S. A. and Huberman, B. A. (2006). The structure of collaborative tagging systems. *Journal of Information Science* 32 (2), 198-208.

[5] Green, R. (1995**).** Topical relevance relationships. I. Why topic matching fails. *Journal of the American Society for Information Science* 46 (9), 646-653.

[6] Heymann, P., Koutrika, G., and Garcia-Molina, H. (2007). Can social bookmarking improve web search? Technical Report InfoLab 2007-33, Department of Computer Science, Stanford University, Stanford, CA, USA, DOI= http://dbpubs.stanford.edu:8090/pub/2007-33

[7] Hutchins, W. J., 1977 On the problem of 'aboutness' in document analysis, *Journal of Informatics,* vol.1, no.1, 17-35.

[8] Khoo, C. S. G. and Na, J. C. (2006). Semantic relations in information science. *Annual Review of Information Science and Technology* 40 (2006), 157-228.

[9] Quintarelli, E. R., Resmini, A. and Rosati, L. (2007). Face tag: Integrating bottom-up and top-down classification in a social tagging system. *Bulletin of the American Society for Information Science and Technology* 33(5), 10-15.

[10] Smith, Gene. (2008). Tagging – *people powered metadata for the social web.* New Riders, CA.

[11] Trant, J. and Wayman, B. (2006). Investigating social tagging and folksonomy in art museums with steve.museum, DOI=http://www.archimuse.com/research/www2006-tagging-steve.pdf

[12] Tuominen, K., Talja, S. and Savolainen R. (2003). Multiperspective digital libraries: The implications of constructionism for the development of digital libraries**.** *Journal of the American Society for Information Science and Technology* 54 (6), 561-569.

[13] Yoon, K. (2007). A study of interpersonal information seeking: the role of topic and comment in the articulation of certainty and uncertainty of information need. *Information Research***,** 12**(**2) paper 304. DOI= http://InformationR.net/ir/12-2/paper304.html

[14] Yoon, K. (2002). Ph. D. Dissertation, School of Information Studies, Syracuse University

[15] Yoon, K. and Nilan, M. S. (1999). Toward a reconceptualization of information seeking research: focus on the exchange of meaning. *Information Processing and Management* 35, 871-890.

## Appendix

| example | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Title | Simplicity is highly overrated | As we may think | Annals of Medicine: The checklist | Is Google making us stupid? |
| Number of Bookmarks | 450 | 1187 | 409 | 4500 |
| Number of Notes | 114 | 351 | 98 | 1146 |
| Top Tags with counts | Design 275 | History 433 | Medicine 172 | Google 2056 |
| | Simplicity 225 | Technology 384 | Health 126 | Internet 1450 |
| | Usability 215 | Internet 315 | Science 88 | Technology 1248 |
| | Marketing 105 | Hypertext 286 | Checklist 80 | Culture 1080 |
| | Articles 57 | Vennervarbush 280 | Productivity 80 | Reading 1001 |
| | Development 45 | Memex 260 | Article 91 | Brain 861 |
| | Interesting 29 | Article 195 | Interesting 48 | Articles 578 |
| | Technology 28 | Science 139 | Toread 46 | Society 426 |
| | Ui 28 | Information 103 | Newyorker 38 | Web2.0 404 |
| | Blog 26 | Web 95 | Process 32 | Education 380 |
| | Culture 25 | Research 76 | Healthcare 24 | Psychology 349 |
| | Psychology 21 | Reference 72 | Research 16 | Web 338 |
| | Ux 21 | Bush 63 | Medical 16 | Research 291 |
| | Norman 17 | Computer 58 | Innovation 16 | Thinking 238 |
| | Complexity 17 | Knowledge 51 | Organization 15 | Media 236 |
| | Software 16 | Culture 51 | Learning 14 | Writing 230 |
| | Hci 15 | Vannevar 50 | Management 14 | Learning 216 |
| | Interface 12 | Web2.0 48 | Gtd 13 | Trends 204 |
| | Interaction 12 | Future 47 | Engineering 13 | Blog 170 |
| | Experience 11 | Philosophy 43 | Blog 13 | Information 170 |
| | Features 9 | Vannevar_bush 40 | Information 12 | Toread 169 |
| | Product 8 | Vannevar-bush 40 | Business 12 | Science 162 |
| | Consumer 7 | Articles 36 | Checklists 10 | Articles 162 |
| | Donnorman 7 | Library 33 | Articles 9 | Future 133 |
| | Interaction_design 6 | Computers 32 | Design 9 | Books 133 |
| | People 6 | Media 30 | Quality 8 | Attention 121 |
| | Don_norman 6 | Memory 30 | Projectmanagement 8 | Intelligence 120 |
| | Computers 6 | Ideas 30 | Complexity 8 | Cognition 103 |
| | Essay 6 | Essay 27 | Essay 6 | Mind 92 |
| | | Vannevar+bush 22 | 2007 5 | Atlantic 81 |
| Date collected | Apr. 21 2009 | May 12, 2009 | June 23, 2009 | July 8, 2009 |

**Table 1. Sample top tags used for example analysis**

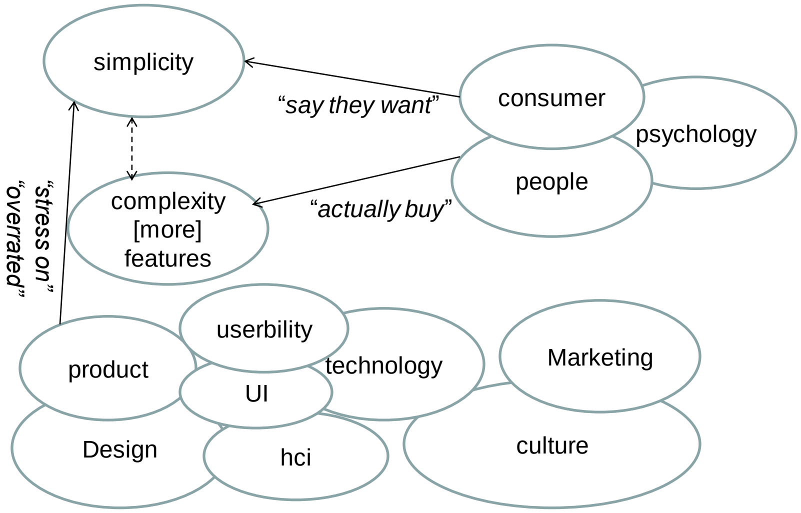


**Figure 1. simple semantic relation of example 1**

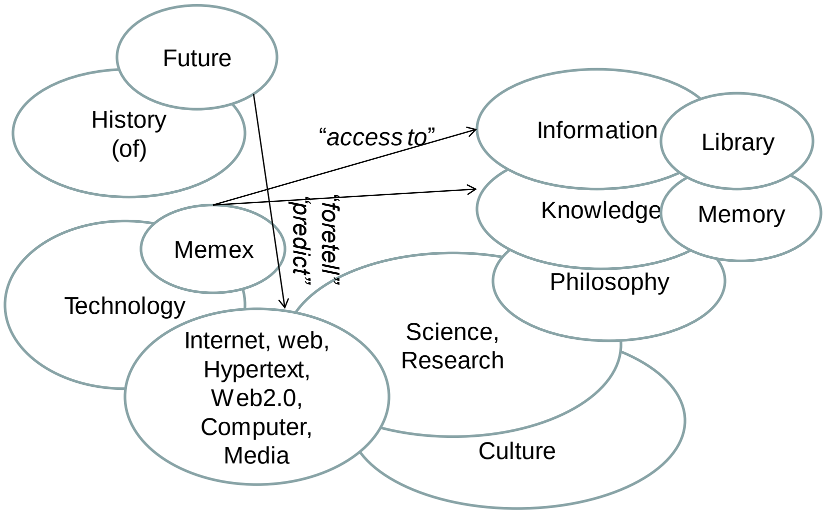


**Figure 2. Simple semantic relation of example 2**

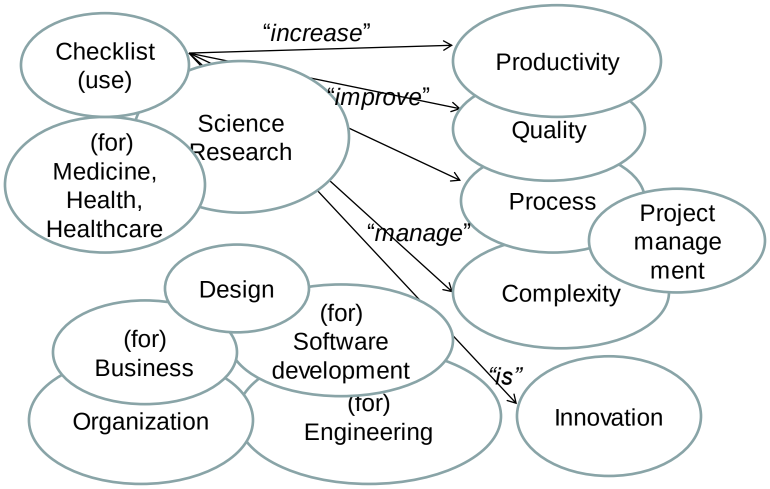


**Figure 3. Simple semantic relation of example 3**

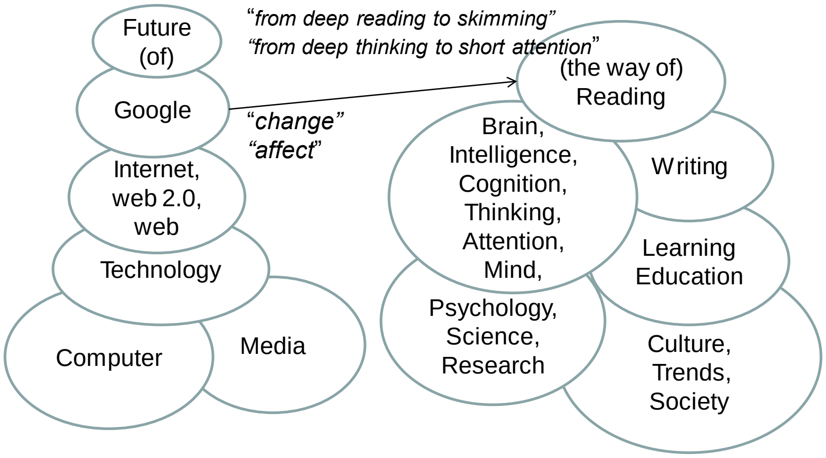

"from deep reading to skimming"
"from deep thinking to short attention"
"change"
"affect"
Future (of)
Google
Internet, web 2.0, web
Technology
Computer
Media
(the way of) Reading
Brain, Intelligence, Cognition, Thinking, Attention, Mind,
Writing
Learning Education
Psychology, Science, Research
Culture, Trends, Society

**Figure 4. Simple semantic relation of example 4**

# Ambiguity and Context-Aware Query Reformulation

Hui Zhang

School of Library and Information Science
Indiana University

hz3@indiana.edu

## ABSTRACT

In this position paper, we suggest that query ambiguity is a major challenge for IR and there is space of improvement for existing approaches. Thus, we propose a novel disambiguation approach that constructs word meanings based on context mining from user sessions in search engine query log. Our preliminary result makes us believe that it is a promising direction. We also discuss how a search interface benefits from this approach in supporting faceted and exploratory search by context-based query reformulation.

## 1. INTRODUCTION

An effective information retrieval (IR) interface should behave similarly to a consultant in a way that it will assist users fulfilling their information needs despite the insufficient state of knowledge. Some progresses have been made towards this goal such as query expansion and relevance feedback in the past several decades. However, the problem of query ambiguity is poorly addressed in previous studies, which is abnormal given its significant impact to human-computer interaction and IR.

There are two major forms of ambiguity in natural language: semantic and syntactic. The source of semantic ambiguity comes from the usage of polysemy[1] whereas the source of syntactic ambiguity comes from the construction of sentences and phrases. Because most of the queries submitted to the IR systems are short [9], polysemy and phrasal structure are prevalent sources of ambiguity for IR. In addition to natural language, another major source of ambiguity comes from user intention. For example, a query such as *DNA* could indicate user's information needs on any of the following topics: health care, law enforcement, or biology.

Most of the researches on query disambiguation rely on existing knowledge resources such as dictionary and WordNet for word sense definitions (e.g., [4, 6, 10]). The limitation of these approaches is that the information contained in the knowledge resources is either inappropriate or outdated for the underlying tasks. Schutze [7] proposed a unsupervised approach by inducing word sense from term clusters. However, the computing cost of this approach is high, which makes it unfeasible for analyzing large collections even with modern hardware.

Provide support to end users on query editing and reformulation is one of the core functions of an IR interface. Many initial user queries are unspecific and incomplete due to anomalous state of knowledge, or *ASK* [2]. A recent study estimated that 16% of the queries submitted to a web search engine are ambiguous [8]. However, because IR system is ineffective on handling query ambiguity, many misleading query suggestions are made. To overcome the limitation of previous methods, we propose solutions for the following three tasks:

1. Establish word meanings by harvesting and clustering query context from user sessions in a query log.
2. Resolve query ambiguity with structured knowledge in Wikipedia and statistical learning.
3. Assist diversity and exploratory search with context-aware query reformulation.

Task 1 is a preparatory task, which can be done off-line. However, tasks 2 and 3 are at heart of a search interface and they have significant impacts on the retrieval performance and user satisfaction. We will discuss issues, methods, and preliminary results for each of the tasks in the sections below.

## 2. Inducing Word Sense with User Session Clustering

Ambiguity of word meaning rises because of the lack of common consent. Therefore, without a universal ontology that provides such common background, one can only establish meanings based on context.

There is growing interest in leveraging user query log for search optimization. However, previous studies focus on association at word level, which ignore the problem of query disambiguation. Consequently, improvement on query disambiguation, if any, is insignificant as a by-product of bag-of-words approach. In contrast, we propose a method that will establish word meanings by harvesting contexts from user sessions in a large query log. In the preliminary study, we defined a user session as all click-through pairs (query and clicked URL) submitted by the same user within thirty minutes. The hypothesis of our approach is that:

- Queries in the same query session normally have similar meanings; therefore, words in those queries can be used to represent one salient sense, or meaning, of a concept that is uttered in the queries.
- One could obtain major senses of a concept by collecting and analyzing its contexts in all query sessions over a large query log.

We also implement an early stage system to verify our approach. The query collection we chose is the AOL query log [5] because it is publicly available. Only queries with clicked URLs are considered and duplicated queries within the same session are condensed. The parsed information is saved in a relational database for later access. Each identified session will be assigned a unique id with information such as user, start time, and end time saved in the database. In addition, each valid click-through pair will be assigned a unique id with information such as query string, clicked URL, and time saved. The relationship between session and click-through is one-to-many. We have so far processed half million lines of user queries to for evaluation. Using the *DNA* example discussed at the beginning of the paper, we could extract its contexts from user session  shown in table 1 (only a portion of the total record is displayed to save space).

[1] One word has multiple possible meanings

**Table 1. Context of word *DNA* extracted from a query log**

| Session_id | Context1 | Context2 | Context3 |
|---|---|---|---|
| 2849961 | maternity | pregnant | test |
| 2949075 | enzymes | restriction | physical property |
| 2949076 | hydration | concentration | spectrophoto meter |

From the table, we can conclude that context words in the first session demonstrate the health care topic of *DNA* whereas context words in the next two sessions probably establish the biology topic of *DNA*. Another finding of the experiment is that sessions are sparse, which means a large number of them only have one or two unique click-through pairs. In the future research, we will cluster sessions based on the click-through information to obtain distinct meanings. Finally, we could map induced word senses to categories of existing knowledge bases such as Wikipedia. The Wikipedia mapping will assign induced word senses with appropriate labels when, and will make the content of Wikipedia usable for query expansion.

## 3. Query Disambiguation with Supervised Learning

User query disambiguation is difficult because the context in the query is inadequate to establish meanings. To overcome this problem, the general strategy of query disambiguation is to develop more effective disambiguation algorithm and adopt features that are more representative. Previous studies suggest that supervised learning approach produces the best disambiguation performance [1]. However, its effectiveness is restricted by the cost of building training data. Popular learning algorithms require hundreds of labeled instances for each sense to make reliable prediction on incoming queries. Given the number of words and the variety of word usage in reality, it is impossible to build the training set manually. Our query log mining approach provides a solution to this bottleneck. At one hand, it is automatic, which reduces the human involvement to minimum. At the other hand, our approach is more efficient and effective than clustering as it takes advantages of crowd intelligence contained in the query log.

Another issue of supervised learning is to extract the most representative features for the underlying decision. For query disambiguation, we propose syntactic features including the part of speech of context and the ambiguous word, identified phrase in the query, position and distance of surrounding context to the ambiguous word.

In the situation that the incoming query contains novel terms that the learning algorithm cannot make a confident decision, we will seek help from the richer context about the query topic contained inside the underlying collection. As this particular operation can be integrated with post-retrieval procedure, we will discuss what it means to the search interface together with query reformulation in the next section.

## 4. Context-Aware Query Reformulation

The goal of query reformulation is modifying the initial user query to make it more descriptive and specific on user's information need. Previous studies on query reformulation primarily focused on synonym problem, or how to replace a word in the query with more descriptive one [3, 11]. In contrast, we attempt to address the problem of query disambiguation by emphasizing the importance of context in query reformulation. There are two types of contexts involved in query reformulation: the first type is referred as query context that includes query terms, user intention, and statistics of the query; the second type of the context is referred as document context, which includes surrounding words that are frequently co-occur with the target word in the top-ranked returns.

Xu and Croft [12] also conduct co-occurrence analysis on query terms in top-ranked documents. Our approach is different because it considers syntactic relationship (e.g., head-modifier, subject-verb, and verb-argument) in addition to statistical data as we believe that syntactic patterns are reliable evidence of meanings. As shown in figure 1, we demonstrate the process of using document context to disambiguate the meaning of *marine* in the query of *marine vegetation*.

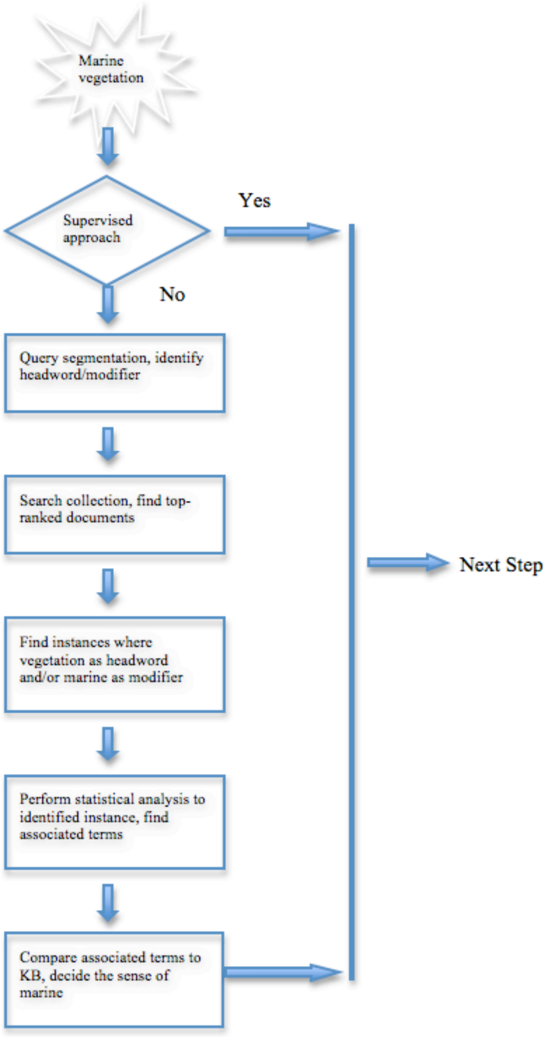

Marine vegetation
Supervised approach
Yes
No
Query segmentation, identify headword/modifier
Search collection, find top-ranked documents
Find instances where vegetation as headword and/or marine as modifier
Perform statistical analysis to identified instance, find associated terms
Compare associated terms to KB, decide the sense of marine
Next Step

**Figure 1. Disambiguate algorithm with document context**

To implement our proposed approach, the developer of the search interface should answer the following two questions:

1. How could a search interface take advantage from query disambiguation and reformulation for faceted and exploratory search?
2. How could a search interface assist query disambiguation and reformulation?

The answer to the first question is obvious: the interface may benefit from the disambiguation process in tasks such as query recommendation (e.g., in sponsored search), query suggestion, and result summarization. In addition, our approach will also create a set of pseudo facets of the query topic from local documents that does not rely on any existing knowledge structures, which could be beneficial for rare query. However, the answer to the second question is still unsettled. Users are reluctant to get involved in an interactive IR process due to lack of motivation or frustration of the interface. An effective search interface should be explicit and engaging. For instance, it could highlight words that are salient to meanings in search snippets. In another example, when the system is unclear about user's intention or the meaning of a polysemy, it could present the most representative result for each candidate to the user in a group with contexts highlighted in the snippet, and let user decide which result to select and learn from that decision as relevance feedback.

## 5. Conclusions

Access to modern IR systems such as web search engine is not limited to professionals any more. In consequence, end users will expect assistance from the system to remedy their lack of knowledge. At the one hand, they prefer to view only a small number of relevant results; but at the other hand, they also anticipate that the system will provide them different perspectives on the topic to complement their initial search. To meet this requirement, we propose methods that will predict user intention and resolve query ambiguity for effective query reformulation. The proposed approach takes advantages of various resources such as query log Wikipedia. The preliminary result makes us believe that it is a promising direction.

## 6. REFERENCES

[1] Agirre, E. and Edmonds, P. Word Sense Disambiguation: Algorithms and Applications. Springer, 2007.

[2] Belkin, N. J., Oddy, R. N. and Brooks, H. M. ASK for information retrieval: Part I. Background and theory. Journal of Documentation, 38, 2 (1982), 61-71.

[3] Guo, J., Xu, G., Li, H. and Cheng, X. A unified and discriminative model for query refinement. In Proceedings of the Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval (Singapore, 2008). ACM.

[4] Liu, S., Yu, C. and Meng, W. Word sense disambiguation in queries. Proceedings of the 14th ACM international conference on Information and knowledge management (2005), 525-532.

[5] Pass, G., Chowdhury, A. and Torgeson, C. A picture of search. ACM New York, NY, USA, City, 2006.

[6] Sanderson, M. Word sense disambiguation and information retrieval. Springer-Verlag New York, Inc. New York, NY, USA, 1994.

[7] Schutze, H. and Pedersen, J. Information retrieval based on word senses. Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval (1995), 161-175.

[8] Song, R., Luo, Z., Wen, J.-R., Yu, Y. and Hon, H.-W. Identifying ambiguous queries in web search. In Proceedings of the Proceedings of the 16th international conference on World Wide Web (Banff, Alberta, Canada, 2007). ACM.

[9] Spink, A., Wolfram, D., Jansen, M. B. J. and Saracevic, T. Searching the web: The public and their queries. Journal of the American Society for Information Science and Technology, 52, 3 (2001), 226-234.

[10] Voorhees, E. M. Using WordNet to disambiguate word senses for text retrieval. Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval (1993), 171-180.

[11] Wang, X. and Zhai, C. Mining term association patterns from search logs for effective query reformulation. In Proceedings of the Proceeding of the 17th ACM conference on Information and knowledge management (Napa Valley, California, USA, 2008). ACM.

[12] Xu, J. and Croft, W. B. Query expansion using local and global document analysis. Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval (1996), 4-11.