HCIR 2008

Proceedings of the Second

Workshop on Human-Computer Interaction and Information Retrieval

http://research.microsoft.com/~ryenw/hcir2008/

Microsoft Research, Redmond, Washington, USA

October 23, 2008

Workshop Chairs:

Daniel Tunkelang, Endeca
Ryen White, Microsoft Research

Program Chair:

Bill Kules, The Catholic University of America

Supporters:

Microsoft Research
Endeca

Second Workshop on Human-Computer Interaction and Information Retrieval

As our lives become ever more digital, we face the difficult task of navigating the complex information spaces we create. The fields of Human-Computer Interaction (HCI) and Information Retrieval (IR) have both developed innovative techniques to address this challenge, but their insights have to date often failed to cross disciplinary borders. In this one-day workshop we hope to explore the advances each domain can bring to the other. Following the success of the HCIR 2007 workshop, co-hosted by MIT and Endeca, we are once again bringing together academics, industrial researchers, and practitioners for a discussion of this important topic.

This year the workshop is focused on the design, implementation, and evaluation of search interfaces. We are particularly interested in interfaces that support complex and exploratory search tasks.

Program Committee:

James Allan, University of Massachusetts, USA
Peter Anick, Yahoo!, USA
Peter Bailey, Live Search, USA
Peter Brusilovsky, University of Pittsburgh, USA
Pia Borlund, Royal School of Library and Information Science, Denmark
Robert Capra, University of North Carolina at Chapel Hill, USA
Ed Chi, Palo Alto Research Center (PARC), USA
Ed Cutrell, Microsoft Research, USA
Ed Fox, Virginia Tech, USA
Gene Golovchinsky, FX Palo Alto Laboratory, USA
Marti Hearst, University of California at Berkeley, USA
Diane Kelly, University of North Carolina at Chapel Hill, USA
Jim Jansen, Pennsylvania State University, USA
Gary Marchionini, University of North Carolina at Chapel Hill, USA
Merrie Morris, Microsoft Research, USA
Jeremy Pickens, FX Palo Alto Laboratory, USA
Yan Qu, University of Maryland at College Park, USA
Amanda Spink, Queensland University of Technology, Australia
Elaine Toms, Dalhousie University, Canada
Martin Wattenberg, IBM Research, USA
Ross Wilkinson, CSIRO, Australia

Table of Contents

Presented Papers

Poster Papers

Helping Users Provide Explicit Context-aware Feedback To Measure Search Experience Satisfaction

Raman Chandrasekar
Microsoft Research
One Microsoft Way
Redmond WA 98052
USA
ramanc@microsoft.com

Matthew R. Scott
Microsoft Research Asia
49, Zhichun Road,
Haidian District
Beijing 100190,China
mrscott@microsoft.com

Dean Slawson
Microsoft Research Asia
49, Zhichun Road,
Haidian District
Beijing 100190,China
deansl@microsoft.com

A.R.D. Rajan
New York University
721 Broadway 4th Fl.
New York NY10003
USA
dra247@nyu.edu

Daniel Makoski
Microsoft Corporation
One Microsoft Way
Redmond WA 98052
USA
danmak@microsoft.com

ABSTRACT

There are many reasons to evaluate the goodness of search engines. We take a quick look at some measures of goodness used today, and list requirements for an additional metric that goes beyond these. We present the Search Experience Satisfaction (SES) metric as a vital addition, filling an evaluation niche, to be used along with result-quality methods (which provide a basic goodness measure) and implicit measures (which provide a sense of user satisfaction without necessarily identifying the causes of satisfaction or dissatisfaction).

We describe a prototype that makes it easy for users to provide explicit feedback without taking them away from their tasks. A light-weight, 'always available' feedback bar is used to collect such feedback along with the user's context. This can be used to compute an SES metric with subscores that help diagnose specific issues or identify desirable features. We present findings from a user study conducted with this prototype.

Categories and Subject Descriptors

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

General Terms

Measurement, Design, Experimentation, Human Factors.

Keywords

Search experience satisfaction, feedback bar, explicit feedback, implicit feedback

1. SES: Why we need an additional metric

Internet and intranet search engines are becoming an integral part of our everyday lives. It is important to evaluate the "goodness" of search engines to help improve the search experience of users and advertisers, and build traffic and revenue.

One way to measure the goodness of search engines is to use result-relevance metrics such as Normalized Discounted Cumulative Gain (NDCG) [3]. These metrics compare search engine results with a gold standard created using human judges. These measures provide an indicator of result quality, and are useful in judging the utility of changes made to the engine, for example in result ranking, but they do not attempt to evaluate user interface (UI) or user experience (UX) features.

Clearly, search engine results pages (SERP) are more than just the list of URLs returned for a query, and there is more to search than finding the perfect site. Complex finding tasks and exploratory queries require visits to multiple sites. Search interfaces integrate information from a number of sources, including news, image and video results, and provide UX features like spelling suggestions, query suggestions, advertisements, 'instant answers', query-class--specific page layouts, cached pages and related pages.

Implicit feedback has been used to evaluate such interfaces, supplementing result-relevance metrics. Fox et al. [2] used an instrumented browser to collect implicit and explicit user satisfaction data such as click-through rates, time to first click, time spent on the SERP and destination pages, how the search session was exited, page and session satisfaction ratings, etc. They then modeled the relationship between implicit and explicit features to predict overall satisfaction from implicit measures.

To improve the user experience, search engines also try out new features, for example Live Search's 'video play on hover". It is not easy to evaluate UI/UX features like these using available measures. To cover the entirety of the user's search experience (including not just perceptions of result quality, but also interface and interaction features that help the user go from intent to task completion), we propose a Search Experience Satisfaction (SES) measure. We expect other metrics of interest, such as traffic, clicks and revenue, to be related to the SES measure.

1.1 Requirements for a SES Metric

We studied a number of metrics that look at the goodness of search, and identified positive aspects and shortcomings of these measures. We then came up with a set of requirements we need in a metric to evaluate search experience:

1. **Reliable**: the metric must show stability across repeated observations and across different observers.
2. **Repeatable**: the methodology should be clearly documented and reproducible by others.
3. **Valid**: the measure must reflect users' real feelings.
4. **Low cost**: it must be relatively low cost to evaluate.
5. **Easy to Interpret**: Goodness measures are useful not just to measure features and systems as a basis for making improvements; they also serve as goals for search engine developers. So the measure should be easy to understand, interpret and explain.
6. **Comprehensive, yet contextual:** The measure should be comprehensive and expressed as one number, but with sufficient detail and granularity of context to help in diagnostics of features and feature components**.**
7. **Generalizable**: The metric must be easy to apply across markets, geographies, languages, time etc., with enough flexibility to evaluate a range of scenarios.

8. **Ability to grade ourselves as well the competition:** the metric should permit us grade not just our own search engines, but also our competitors' engines.
9. **Scalable:** the metric should be able to handle features that affect millions of users**.**

Result-relevance metrics such as NDCG do not attempt to evaluate user interaction. Implicit measures capture some aspects of user satisfaction, and when explicit measures are modeled with implicit information, they satisfy several of these requirements. However, they typically do not have context for diagnostics of specific features or components.

Explicit feedback from users can satisfy many of the requirements above. They can be easy to interpret, comprehensive while incorporating detail, easy to apply to the competition, scalable and generalizable. They can be the basis for a SES metric. The challenge is to define a simple mechanism and a methodology that encourages users to provide explicit feedback, and then to show that the metric we compute from users' explicit feedback is reliable, repeatable and valid.

In the rest of this paper, we list design goals for such a feedback system. Based on these goals, we propose a light-weight mechanism to gather explicit user feedback on the entire experience of using a search engine. In particular, we use a feedback bar with 'smiley' icons which the user clicks on to provide feedback; we record the feedback along with the user's current context. We describe an implementation of the feedback bar, and detail a user study that was conducted with this implementation. This feedback can be sliced and diced in many ways to analyze the search experience.

2. THE SES FEEDBACK MECHANISM

In this section, we discuss design goals for a feedback mechanism, and describe a mechanism and a methodology to garner explicit feedback on SES.

2.1 Design Goals

A good feedback mechanism should encourage feedback clicks but discourage click-spam. It should appear serious and legitimate without appearing boring, or, at the other extreme, looking like a flashy advertisement. Such a feedback mechanism should be:

1. Noticeable but not intrusive; easy to switch off/ignore
2. Easy to add to a web page
3. Lightweight, i.e. require very little additional bandwidth
4. Fast and very responsive
5. Easy to use to provide feedback
6. Functional, but not too staid or boring; nor flashy like an advertisement
7. Intuitive, and easy to interpret and use
8. Neutral, and not bias users in any way
9. Designed to be consistent with overall web page theme, for a range of pages
10. Small in size, using very little screen real-estate; it should not take the user away from the task at hand.

2.2 An SES Feedback Bar Prototype

Based on the requirements and the design goals above, we decided on using a feedback bar which is always available on the user's screen. Users are encouraged to provide explicit feedback on search tasks and search result experiences using, minimally, simple clicks on the feedback bar. They can optionally provide task data and verbose feedback. We collect clicks, task data, user context and time, and generate metrics and actionable reports.

This feedback bar is introduced when the user comes to the SERP, say, for example, the Live Search results page (Fig. 1).

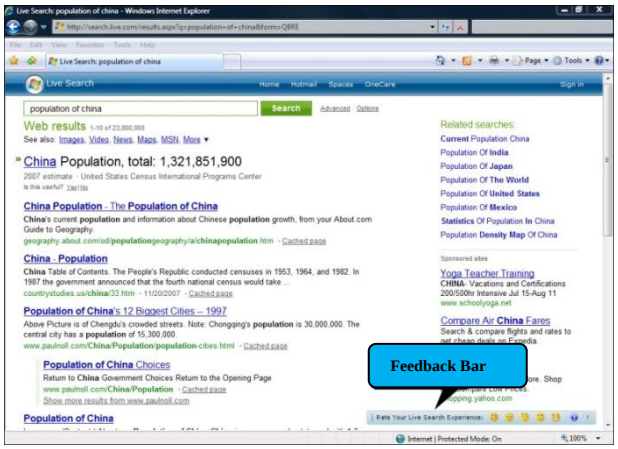


Figure 1. The SES Feedback bar on a SERP

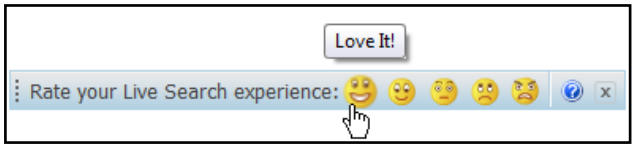


Figure 2. The SES Feedback Bar

Fig. 2 provides a close up of the feedback bar. There are 5 levels of satisfaction that the user can choose, on a Likert-like scale, ranging from **Love it** (value = 5) to **Hate it** (value = 1). Although Likert scales are more typically used to assess agreement or disagreement, we use it here as a way of expressing user satisfaction.

The feedback bar is positioned in the lower right of the screen by default, and floats there on top of the page even as the page is scrolled. The user has the option of repositioning the bar anywhere on the page. If the user clicks on one of the smiley icons on the smile bar, the user is shown a tell us a tiny bit more link (Fig. 3) and given the option to provide more feedback. *Note that we are happy even if the user just clicks on a smiley, giving that basic feedback. Also, unlike several feedback systems, we accept both positive and negative feedback.*

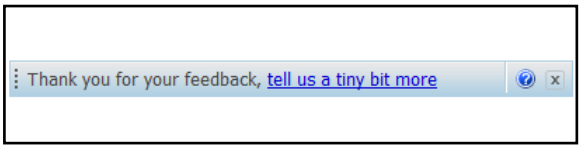


Figure 3. The SES feedback bar showing the *tell us more* link

If the user then clicks on the tell us a tiny bit more link, a feedback box opens up (Fig. 4) for the user to provide information about the query type and any textual feedback he/she may provide. *Unlike many feedback systems which interrupt the user's task context and take them to a new feedback page in their browser*, this feedback box is displayed close to the initiating mouse click, without disturbing the user's context. This preserves

the feedback momentum. If the user does not click on the tell us a tiny bit more link in a few seconds, the feedback bar reverts to the smiley icons shown in Fig. 2.

The query types are based on an extended version of the query classes proposed by Broder [1]. People's search behavior depends on the query type. For example, informational queries may have many clicks on a search result page, while question-answering queries may not elicit any clicks at all. So it is useful to gather this information.

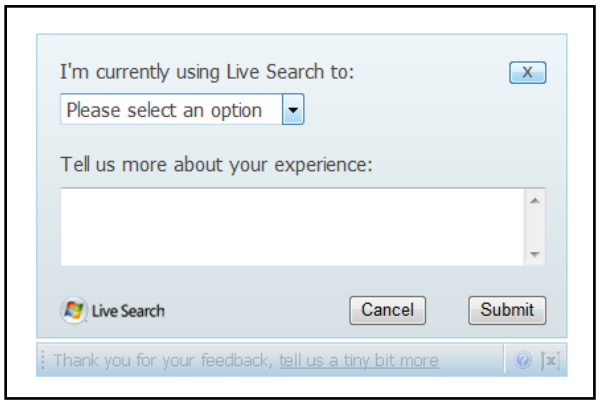


Figure 4. The expanded SES feedback bar

The user can provide further feedback on this or other result pages. The feedback data collected from this mechanism is used to compute a series of user experience metrics.

2.3 Feedback Methodology

Using a simple process, any web page, and in particular search engine result pages (SERPs) can be altered to display the floating feedback bar. Randomly sampled users could be shown the feedback bar. The sampling must be large enough for meaningful statistics, but small enough that spamming in this set will not be cost-effective.

The user can click on a smiley at any point in the search process, whenever a page with the feedback bar is displayed. Every time the user clicks on a smiley, we collect the following data:

- Anonymized User Id
- Page context including the URL or other information such as Query, Market, Form code etc.
- Time
- Satisfaction level (which smiley was clicked)

If the user clicks on the 'tell us a tiny bit more' link, we also collect the query type and/or verbose feedback. From this data, we can devise and compute a Search Experience Satisfaction (SES) score. For example, this can be a number between 1 and 5 (higher is better), composed of weighted SES subscores across query types, as well as SES subscores for each query type, all weighted by time, and calculated differently for each query type. The SES score becomes the major metric to track and improve upon.

Unlike most available feedback systems, the user is not constrained to one feedback item per session or page. The user can submit more than one click for any page; this is very useful, helping us evaluate more than one feature on a page.

We can use this system to compute and compare SES score values for control and experiment pages. We can also pivot on query types, markets, time of day etc., to determine which features perform well, and which do not. For example, if we pivot on query type, as shown in Fig. 5 (based on made-up data), we may infer that we need to improve results for our navigational queries (labeled here as "Get to a specific website").

Overall Averages by Query Type

Experiment Id	Query Type	Total Smile Clicks	Overall Avg SES Value
ex888	Do something else	1107	4.33
ex888	Find specific information	4251	4.52
ex888	Get to a specific website	3105	3.47
ex888	Just a Smiley	9525	3.95
ex888	Surf the web	2207	4.21
ex888	Unspecified	1036	3.58

Figure 5. Sample SES metric report

We optionally accept verbose feedback from users, and we could use text mining on this to gain product insights. Finally, we could model SES as a function of other metrics (like click-through rate, NDCG, etc).

3. FEEDBACK BAR: USER RESEARCH

This section describes some user research we conducted on a prototype of the SES feedback bar. While the user research was done using the Live Search engine, the results are applicable to other search engines.

3.1 User Research Objectives

We had the following user research objectives:

1. Test if feedback from the feedback bar correlates with verbal feedback on Live Search experience.
2. Get users' reactions to our current design of the feedback bar.
3. Test whether user interaction with the feedback bar matches the interaction flow we designed for the bar.

We had 8 participants in our study, all fluent English speakers, 5 male and 3 female, in the age group 17-35 years. All of them used the internet and search engines on a daily basis. The engines they used were primarily Google and Live Search; they also used Yahoo!, Ask and Wikipedia.

Every user was presented with 2 sets of 5 tasks, each consisting of 2 web search based tasks, 1 new search based task, 1 image search based task and 1 video search based task. The feedback bar was displayed in these sets to users in a balanced manner.

The usability engineer observed and recorded how the participants reacted with and used the feedback bar. The engineer also got verbal feedback on search experience (on a 1 to 5 scale, where 1 is very bad and 5 is great) for the set of tasks where the user did not have the feedback bar. The user was also asked several questions on their perceptions and use of the feedback bar.

3.2 Findings and Insights

Here are some findings and observations from the user research.

3.2.1 Validity of the feedback bar metrics

The ratings gathered through the feedback bar and through verbal feedback were very close, as seen in Fig. 6 and Fig. 7. The blue lines refer to tasks for which we got verbal feedback and the red

ones where we got feedback through the feedback bar. The numbers A1, A2, B1, B2 etc. refer to specific tasks. For the majority of the tasks (9 out of 10), the difference in ratings between the two forms of feedback does not exceed 0.5.

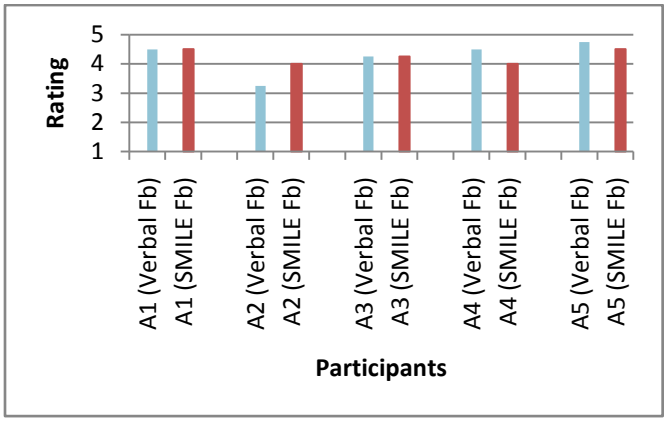


Figure 6. Average rating for tasks in Set A with and without the feedback bar.

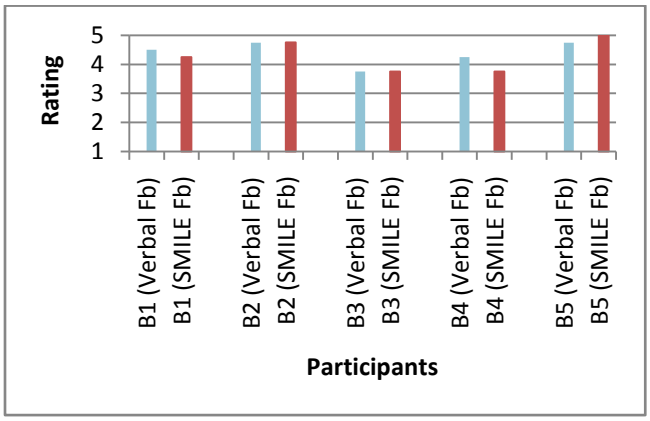


Figure 7. Average rating for tasks in Set B with and without the feedback bar

3.2.2 Users' feedback strategies and concerns

Six out of 8 of the participants had given feedback to some service at some point in time, but only when their experiences were extremely good or extremely bad. Feedback is not a priority and only given when the person is casually browsing the Internet. While performing an important task, the participants would only give feedback if they deemed something truly bad.

Three participants did not normally give feedback because they felt that their feedback does not improve their online experience. One participant does not give feedback because he thought his effort would be exploited for commercial purposes.

In the version we tested, the link shown in Fig. 3 read as 'tell us more'. Four out of 8 people expressed the desire to click on this link. However, all 8 participants felt that this would take them to a page containing a number of questions to be answered. Learning from this study, we changed the link to tell us a tiny bit more to indicate that we did not expect tomes of feedback.

3.2.3 Visibility and perception of the feedback bar

Five out 8 people noticed the feedback bar. However, they only glanced at it and skipped the accompanying text. They perceived the bar as a pop-up or an advertisement when they saw it the first time. Based on this feedback, in the final version, we added a message that is shown once to each user, to tell people about the feedback bar.

Seven out of 8 participants preferred the smileys over other feedback mechanisms like stars, thumbs or numbers. One participant said "They (the smileys) transcend cultural boundaries" and another called the design "very solid and professional."

Seven out of 8 participants were happy with the current design of the feedback bar. Five people noticed the bar at the first instance and 4 people tried to place it in a different position on the page.

Four out of 8 people felt that 5 smileys are the perfect number and express the correct range of options. One suggested an even number of smileys and another said 3 smileys are sufficient.

All 8 participants were fine with the order of the smileys i.e. from happy face to angry face (left to right).

4. DISCUSSION

The focus of this paper is on developing and user testing an always available mechanism to collect explicit feedback that can then be used to develop a new metric covering search experience satisfaction. We identified requirements for the new metric and the design goals for a mechanism. The user research validated some of our ideas and highlighted areas where we could improve.

As the user research points out, there are a number of changes that we can try out. For example, one option worth exploring is to make the feedback bar an integral part of the SERP, rather than have it be a floating bar. Another issue: when we save users' feedback, we currently also save page context with information extracted from the parameterized URL. Will saving the whole page, including advertisements, give us greater feature coverage?

The user research described here is on a small sample of 8 people. The next step is to deploy this feedback bar in a live system, and evaluate actual use. Deployment will tell us if our lightweight bar encourages users to give feedback, especially multiple times on a single page. Further work is required to define and tweak a sensible SES metric from the data collected, and to test the reliability and validity of the metric. We can also extend this methodology, with changes, to applications other than search.

5. ACKNOWLEDGMENTS

This work was done when the authors were all at Microsoft Research Asia in Beijing. We thank all the participants in these tests and the reviewers for their insightful comments.

6. REFERENCES

[1] Broder, A. 2002. A taxonomy of web search. ACM SIGIR Forum, 36(2), Sep 2002.

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

[3] Jarvelin, K., Kekalainen, J. 2002. Cumulated gain-based evaluation of IR techniques. ACM Trans on Information Systems, 20(4), 422-446.

Polestar: Assisted Navigation for Exploring Multi-dimensional Information Spaces

Davor Čubranić
Business Objects, an SAP company
910 Mainland Street
Vancouver, BC, Canada
davor.cubranic@sap.com

ABSTRACT

We describe a system for interactive exploration of multi-dimensional information spaces with which user may be relatively unfamiliar. Our tool, named Polestar, assists the user through a novel combination of several techniques: guided faceted browsing, multiple summarization perspectives of data in the information space during the navigation, and a flexible interaction model that provides both an overview of available choices at each navigation step and an intuitive interface for navigation. We report the details of each of these three components and how they are used for interactive exploration of business intelligence (BI) content.

Categories and Subject Descriptors

H5.2 [**Information interfaces and presentation**]: User Interfaces—*Graphical user interfaces*

General Terms

Design, Human Factors

Keywords

Interactive data exploration, faceted browsing, navigation

1. INTRODUCTION

Pervasive use of information technology in modern enterprises has resulted in large volumes of data that can be used by an organization to better understand, analyze, and even predict what is occurring within and around it, in its environment. Turning this flood of data into useful information, and then delivering and presenting it to the relevant members of the organization is achieved by an array of technologies, applications, and processes collectively commonly called *business intelligence* (BI) or *visual analytics*.

Business intelligence technologies like OLAP give business analysts the capability to digest and understand large volumes of information organized in complex, multidimensional spaces. The tools that these analysts use come with suitably complex and powerful interfaces, such as a pivot table, since they are primarily used by experienced users who know the tools and their data well. However, a large percentage of those using business intelligence [1] solely *consume* it. Those users may have need to perform their own analyses occasionally, but are hampered by their unfamiliarity with the information spaces. For those users, getting an overview of a complex multidimensional space, not to mention finding a small subspace that contains the information that they are looking for, is difficult.

These "casual analysts" need tools that allow *exploration* of data without advance knowledge of its schema. They do not have such tools today; however, a technique for exploratory navigation of structured, multi-dimensional data sets already exists in the knowledge management and information architecture communities: *faceted browsing*.[6] In faceted browsing, the information—such as books in a library or products in a sales catalogue—is classified along multiple orthogonal dimensions of the data, called *facets*. The user browses the information collection by selecting values in facets, often through a simple point-and-click interface. The selected values act as constraints on a view of items in the collection, narrowing it to view only the items that have the selected values in their facets

The advantage of faceted browsing over keyword searching or writing database queries is that the user can always see the available options for constraint values, thereby avoiding empty result sets of a query or the feeling of being lost in an unknown dataset. In this paper, we describe a tool, named Polestar, in which we apply the principles of faceted browsing to the "casual BI analyst" scenario described earlier. Polestar employs a novel user interface for faceted browsing and combines it with a model of exploration utility to help the user make good navigation choices while browsing the information space. Furthermore, because BI data on which Polestar is used revolve around numeric *measures*—unlike catalogue-type collections commonly seen in faceted browsing systems—Polestar also provides graphical summaries of these measures as an additional navigational aid for the user.

2. POLESTAR

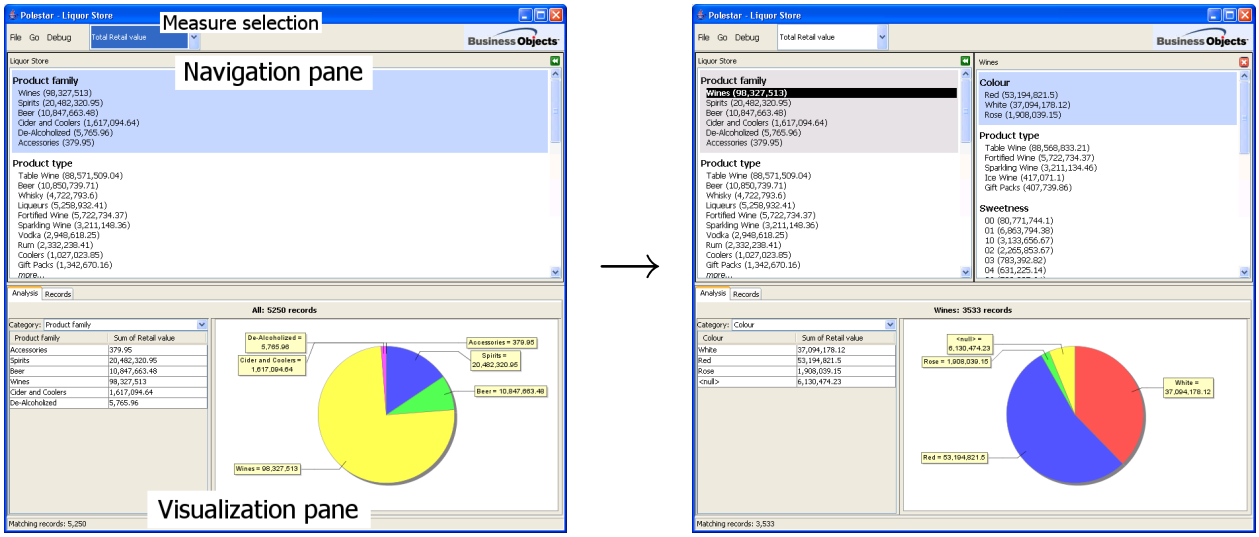
Polestar is a system for interactive exploration of multi-dimensional information spaces with which user may be relatively unfamiliar. The tool assists the user through a novel combination of several techniques: guided faceted browsing, multiple summarization perspectives of data in the information space during the navigation, and a flexible interaction model that provides both an overview of available choices at each navigation step and an intuitive interface for navigation.

2.1 Data Sources

Polestar can use as input any data source that is equivalent to a relation—that is, which contains a set of n-tuples of typed attribute

[1] Approximately 85%, according to Forrester Research.[4]

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
HCIR 2008 Redmond, WA, USA
Copyright held by author.



(a) Initial view

(b) After selecting "Product: Wines"

Figure 1: The Polestar user interface, with the navigation pane in the upper half of the window and a visualization pane showing a summary view in the lower half. The exploration space shown contains inventory and sales data for a fictional liquor store. The screenshots show two points in the exploration: at the very beginning (a) and after selecting "Wines" in category "Product family" (b).

values. The data can come in a variety of formats, such as a plain text file with tuples separated by newlines and attribute values within a tuple by commas or tabs (so-called "CSV files"), a database table, or the result of a join between two or more tables.

The input relation is used to build an *exploration space*. A Polestar exploration space consists of navigable *categories* and numeric *measures* that can be part of mathematical expressions. These components of the exploration space directly correspond to attributes of the relation: non-numeric attributes are turned into categories (subject to having "sufficiently few" distinct values [2] to filter out attributes like unique ids, for example), while numeric columns become measures. Tuples of the relation are also the basic atoms of the exploration space.

2.2 User Interface

The Polestar user interface is composed of two main areas (see Figure 1). In the top half is the navigation pane. The focus of the pane is on a list of categories available for navigation. The list can be ordered alphabetically, by frequency of occurrence, or by the usefulness of the category for navigation (described in the following section). Initially, only the top five categories are shown, but more can be displayed by clicking on the "more" button. Within each category, its available values are shown in the descending order of their frequency in the data set. Again, this list is truncated to show only the first few elements (up to ten, in the current prototype), but can be expanded by the user. Next to each category value is displayed the summary of a user-selected measure at that point in the navigation space, computed using an aggregation function like *sum* or *count*. The measure and function used for this summary are selectable from a drop down menu at the top of the pane.

The lower half of the Polestar window contains the visualization pane. This pane displays a summary view in table and bar chart form of the selected measure aggregated along a user-selected category. Initially, the selected category is the top-ranked category in the navigation pane, but the user can switch to a different one by clicking on the category name in the navigation area. The visualization area thus shows an overview of a measure as seen from the user's current location; furthermore, because the user can change the category along which the measure is being summarized (as well as change the aggregation function, or use a different measure altogether), he or she can see this overview from several different perspectives and gain better understanding of the data.

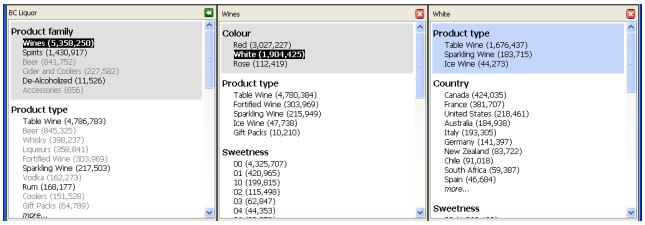
Lastly, the visualization pane can show a "raw" view of the records in the exploration space, accessible under the "Records" tab, for inspection of individual items in the dataset.

Navigation: Similarly to other systems for faceted navigation, a category value is selected by clicking on it in the navigation area. This selection acts as a dynamic filter [1], so that records that do not contain that value are filtered out of the view. Following the selection, the visualization area is updated according to the new, narrowed-down view of the records in the exploration space. Furthermore, the category values that exist in the new view are recomputed, and the new list of categories displayed in the navigation area.

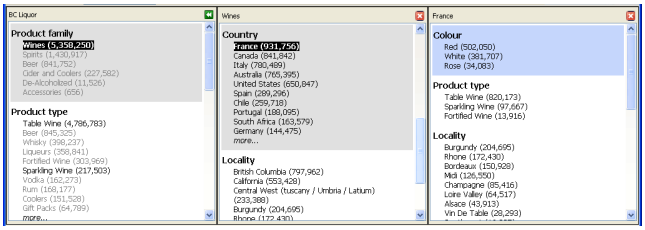
However, this list is shown in a new column to the right of the already existing one (Figure 1(b)), similarly to opening a folder in the NeXT (or Mac OS X) file browser. In addition to the new category listing, the new column also shows updated measure summaries, reflecting—just like the visualization area—the current view of the records.

As more columns are created to their right with each additional navigation step, the existing columns remain unchanged. This visual progression of columns allows the user to maintain an overview not only of the navigation path, like a "breadcrumbs trail" of selections would, but also of the view of the exploration space at each step of the path and choices available for selection. Based on our evaluation of early prototypes of Polestar with information analysts, this overview is crucial to letting the user keep a sense of control and remain oriented in the information space. This find-

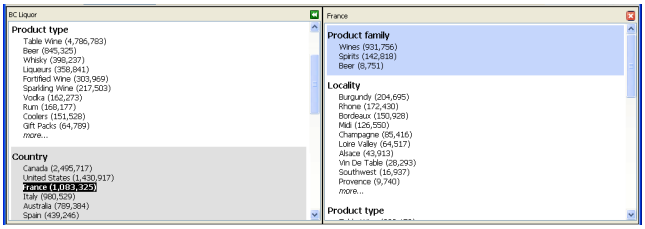
[2] We currently use as threshold 25% of the number of tuples.



(a) "Product: Wine" + "Color: White"



(b) "Product: Wine" + "Country: France"



(c) "Country: France"

Figure 2: Navigation pane as the user adds value "White" in category "Color" to the existing selection (a); changes the selected value in the most recent navigation step from "Color: white" to "Country: France" (b); and removes the second navigation step to set the filter to all products from France (c).

ing is consistent with those of de Alwis and Murphy in their study of disorientation in software development tasks, where they identified "the absence of connecting the navigation context during program exploration" as a significant contributing factor to a sense of getting lost.[3] Our multi-column browser maintains a high *visual momentum* [5] during navigation, unlike the current UIs for faceted navigation in which the display completely changes after each navigation step, isolating it from those before and after and transferring the burden of transitioning and reorienting to the user.

Altering navigation path: Another benefit of the multi-column feature is that exploration of the space becomes very simple: all columns are interactive and clicking on another value in any of them changes the selection at *that* step only—other selections in the navigation path remain the same as long as they do not lead to an empty matching set, which would violate the basic principle of faceted browsing, or are discarded otherwise. Similarly, a navigation step can be removed, thus broadening the record view, by clicking on a column's close button. Figure 2 illustrates this sequence of operations, continuing with our running example: in pane (a), the user adds a new filter to the selection, white wines. In pane (b), the user changes this step from "Color: white" to "Country: France". Lastly, in pane (c), the user closes the second column ("Product: wine") to broaden the view of the space to all products from France.

Importantly, changing the selection in an intermediate navigation step, or removing a step, will maintain the same summary criteria (category of aggregation) in the visualization pane. This way the user can quickly examine two related subspaces, such as French white and red wines in the example above.

2.3 Navigational Utility Model

One of the challenges for the users of faceted browsing systems is that if the data set is very large and contains a lot of facets, and for at least some of those containing many attribute values, navigating through the data efficiently can become very difficult. The difficulty is even greater if the user is not closely familiar with the data and does not know which subset of the data might contain interesting information.

To assist such users navigate an exploration space efficiently, Polestar introduces a model of the usefulness of a category as a selection choice to narrow down the view into the space to a subset of its records. The intuition behind the model is threefold:

1. Partition the available space for efficient navigation
2. Make it easy for the user to select a category value for navigation
3. Avoid focusing prematurely on a small part of the collection

These three components can be expressed directly in terms of measurable properties of the data in the exploration space as follows:

Efficient partitioning: translates into an even distribution of values in the category (entropy);
Easy selection: expressed as a small number of distinct category values (cardinality);
Good coverage: defined as the percentage of non-null values in the category.

Finally, the components need to be combined into a single category score so that the basic properties of the model can be maintained. That is, we want to favour *high* entropy, *low* cardinality, and *high* coverage. The exact calculation is performed as described in the remainder of this section.

Given a category c which consists of a set of values c_i, the entropy of the category $\mathcal{H}_c$ is defined as follows:

$$\mathcal{H}_c = \sum_i p(c_i) \log p(c_i) \tag{1}$$

where $p(c_i)$ is the probability of the category c having a particular value c_i, and is calculated as the frequency (number of occurrences) $f(c_i)$ of the value c_i divided by the sum of frequencies of all values of that category: $p(c_i) = \frac{f(c_i)}{\sum_j f(c_j)}$.

Cardinality $|c|$ of a category c is the number of distinct values in the category, the standard definition of set cardinality in mathematics.

Coverage Z_c of a category c is the proportion of tuples in the data set for which c has non-null value, that is:

$$Z_c = \frac{\sum_i f(c_i)}{N} \tag{2}$$

where $f(c_i)$ is the number of occurrences (frequency) of category value c_i, and N is the total number of tuples.

Finally, we can calculate the score S_c for each category c by combining the values for category's entropy $\mathcal{H}_c$, coverage Z_c, and cardinality $|c|$:

$$S_c = \frac{\mathcal{H}_c Z_c}{|c| log |c|} \tag{3}$$

Of course, as the user navigates by making selections, the navigation space shrinks as more constraints are added to filter out non-

matching tuples. Therefore, the distribution and number of category values changes at each navigation step, and category ranking scores need to be recomputed accordingly.

2.4 Implementation

The version of Polestar described in this paper is implemented as a standalone Java application. It can use as its input data in CSV files, a table in relational databases (accessed as ODBC data sources), and data derived from more complex relational schemas accessed through Business Objects Enterprise platform (so-called *universes*). We are currently rearchitecting the application by splitting it into a core engine that runs as a service within a web application server and provides calculation of category rankings and measure summaries, and a web-based front end that uses Adobe Flex framework for the user interface.

3. RELATED WORK

Since Yee et al.'s 2003 paper [6], faceted browsing has become a common sight in online shopping sites, libraries, and other UIs for searching and exploration of datasets with multiple, orthogonal attributes. Polestar differs from existing faceted browsing systems in two respects: one is in its application, which is not to find a particular item in a catalogue, but to gain an understanding of business intelligence data, that is, ultimately of numbers expressed as measures in our data model. For this reason, the faceted navigation in Polestar works in concert with the visualizations in the summary pane, and its main purpose is to quickly and efficiently navigate the information space so that the user can explore it from multiple viewpoints. Secondly, in such an application maintaining the context and remaining oriented during navigation becomes even more important, which is why we have developed the multi-column browser.

Ranking of data attributes for display and navigation purposes has been investigated by Dakka et al.[2] Their focus is on automatic construction of concept hierarchies from free-form word annotations, or "tags," and the selection of best portions of those hierarchies when the screen space is limited. Their ranking is done on values *within* a facet; when there are multiple dimensions present in the dataset, they are still ordered statically in the user interface. Our focus, on the other hand, is to help the user decide which dimension to use as the next axis of navigation, which is why our rankings are calculated *between* categories.

4. CONCLUSION AND FUTURE WORK

We have shown how faceted browsing is used for interactive exploration of business intelligence (BI) content in the Polestar tool. Polestar provides a flexible and intuitive user interface for faceted browsing, combined with visual summaries of a measure value in the active view of the information space, and assists the user make efficient navigation choices. Feedback gathered from the users during preliminary usability evaluation of Polestar has been extremely positive, and has encouraged us to continue with the development of the tool. Future research includes: extending the ranking model so that the score depends on the shape of measure values, in addition to the distribution of category values, thus identifying regions of unusual data; incorporating hierarchical relationship between categories where it is explicitly defined by the data source; modifying category scores to promote navigation paths commonly taken by other users working on a similar task (collaborative filtering); as well as more extensive usability evaluation.

5. REFERENCES

[1] C. Ahlberg and B. Shneiderman. Visual information seeking: tight coupling of dynamic query filters with starfield displays. In *CHI '94: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 313–317, 1994.

[2] W. Dakka, P. G. Ipeirotis, and K. R. Wood. Automatic construction of multifaceted browsing interfaces. In *Proceedings of the 14th ACM international conference on Information and knowledge management (CIKM)*, pages 768–775, 2005.

[3] B. de Alwis and G. C. Murphy. Using visual momentum to explain disorientation in the Eclipse IDE. In *Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing (VLHCC)*, pages 51–54, 2006.

[4] K. Gile. In search of the ubiquitous analytic end user: Profiling the analytic end user. *DM Review*, Sept. 2003.

[5] D. D. Woods. Visual momentum: a concept to improve the cognitive coupling of person and computer. *Int. J. Man-Mach. Stud.*, 21(3):229–244, 1984.

[6] K.-P. Yee, K. Swearingen, K. Li, and M. Hearst. Faceted metadata for image search and browsing. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI)*, pages 401–408, 2003.

UIs for Faceted Navigation
Recent Advances and Remaining Open Problems

Marti A. Hearst
School of Information
UC Berkeley
Berkeley, CA 94720
hearst@ischool.berkeley.edu

ABSTRACT

Faceted navigation is a proven technique for supporting exploration and discovery within an information collection. The underlying data model is simple enough to make navigation understandable while at the same time rich enough to make navigation flexible in a wide range of domains. Nonetheless, there remain issues in both the presentation of navigation options in the interface and in how to extend the model to allow more flexible discovery while still retaining understandability. This paper explores both of these issues.

1. INTRODUCTION

Faceted navigation is a proven technique for supporting exploration and discovery [8, 4] and has become enormously popular for integrating navigation and search on vertical websites. Its popularity is attested to in part by the fact that content management architectures, such as Solr and Drupal, contain support for faceted navigation. Despite its widespread use, there are design challenges inherent in building the interface for faceted navigation. The two biggest challenges are: (i) poor choices in the design can lead to decreased usability of the interface, and (ii) large category systems, especially subject-oriented category systems, are still not well-supported in the interface. This paper discusses these issues in the context of some recent innovations in the design space for faceted navigation and discusses some future directions.

2. BACKGROUND AND TERMINOLOGY

The starting assumption is that the overall goals of faceted navigation are to support flexible movement through the information space, provide suggestions of navigation choices at each point in the search process, provide seamless integration with keyword search, allow for fluid switching between refining and expanding, prevent empty result sets, and provide a feeling of control and understanding without confusion.

Facets refer to categories used to characterize information items in a collection. A facet can be flat or hierarchical; in either case, a set of labels is associated with each facet. Portions of the hierarchy within a facet is that facet's subhierarchy. In an information collection that supports faceted search, multiple labels are assigned to each item, as opposed to a strictly hierarchical system in which items are placed into single categories or folders. (In this respect, faceted information structures bear some relationship to social annotations, or tagging, that is a popular user-participation form of metadata assignment today. In fact, I believe that tags can provide an excellent basis for the formation of better organized faceted navigation structures, but that is a different topic.)

In the faceted navigation interface, when a label is selected by a user, all items that have been assigned to that label are retrieved, so selecting a label within a facet hierarchy is equivalent to querying on a disjunction over all the labels beneath the selected one. When labels from different parts of the hierarchy are selected, the system in effect builds a conjunct of disjuncts over the selected labels and their subcategories.

In an earlier paper [3], I laid out some issues surrounding the design of faceted interfaces and their interface solutions. In particular, that paper discussed how to clarify navigation within and across facet hierarchies, how to represent history (breadcrumb trails), the importance of incorporating keyword search within the faceted structure, the importance of details in graphic design, and innovations in facet exposure choices as put forward by eBay Express.

In this paper, I extend this discussion to reflect advances that have occurred in the interim, as well as to underscore some of the remaining issues.

3. MIXING CONCEPTS WITHIN FACETS

Faceted navigation generally works best if the facets are conceptually orthogonal and the item assignment is responsible for mixing and matching them. However, there are many cases in which some concepts mix with only a subset of other concepts, and so grouping them in the interface might make the relationships clearer. Getty Images' faceted interface has an interesting way of doing this. Figure 1 shows facets about characteristics of people grouped all in one super-facet. This is similarly done for Style divided into Composition, Viewpoint, and Image Technique. Although conceptually this approach is not different than the standard approach (as seen in Flamenco [8] and many commercial sites), the visual grouping of related but orthogonal modifiers seems like a good idea. Unfortunately, there is a substantial problem with the facet organization in this interface. The grouping called Keywords consists of both Concept and Subject, and these in turn contain a hodgepodge of subject categories. Thus this interface does not address the problem of how to deal with a large number of subject labels.

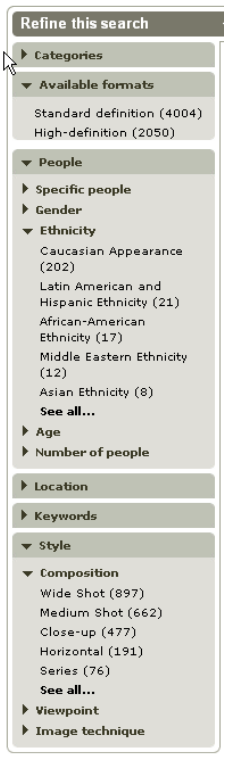


Figure 1: Getty Images' faceted navigation interface uses a graphic design to visually group related facets together.

4. INTEGRATING "SMARTS" INTO SEARCH USER INTERFACES

Aided by support for fast client-side processing, it has become feasible to incorporate information related to the users' query in dynamic, and sometimes subtle ways. Below I discuss two exciting examples of this development as they intersect with faceted navigation interfaces.

4.1 Auto-Suggest Search Within Facets

Auto-suggest, aka auto-compete, aka dynamic term suggestions is a mechanism in which, as a user is typing a query term into the entry box, queries that are lexically related and that have been asked by other searchers in the past are shown beneath the entry form [1]. . This is an attempt to help the user finish formulating their query by showing what should by highly relevant terms, and seems to be a generally a good idea that should be used wherever possible. This is a rare case in which there have been few if any usability studies (the closest to it that I know of is by White and Marchionini [7]), but by observation and anecdote, I am willing to claim that the usability appears to be very high.

A twist on the idea is to provide separate autocomplete entry forms for each facet [2]. This is especially useful for facets with very large numbers of labels that cannot be organized into a hierarchy; a common example is names of authors in a bibliographic collection. But even for facets with fewer labels, dynamic suggestions of terms related to the letters typed so far seems to be a helpful and usable feature.

4.2 Keyword Search Terms Affecting Facet Label Ordering

Before discussing this feature, some background information is needed. As discussed in an earlier paper [3], eBay Express introduced a number of innovations in their method of presenting faceted metadata. Rather than placing the facets on the side, which can require scrolling by the user, they place a small number of facets (four or five) in the interface "sweet spot" across the top of the page, showing only a few labels per facet, and a *More...* link to see the rest. Clicking on this link brings up a dialog box containing checkboxes, allowing the user to create an OR (disjunction) over the choices within one facet. The designers determined in advance (largely through query logs and click logs) which facets are most important for each major product type, and initially expose those facets only, with a compressed list of additional facets on the line below. Selecting a facet adds it to the query representation (the breadcrumb) and causes that facet to disappear from the main canvas, and be replaced by one of those not expanded yet.

Another innovation was to employ cleverness in the handling of keyword queries. A query on "women's rebocks" within the Shoe product space would show the corresponding facets *Type* > *women's* and *Brand* > *Rebock* selected already within the query breadcrumb. This is terrific when it works, of course, but in many cases the mapping might not be correct.

Recently the lifestyle website Yelp converted its navigation interface to eBay Express-style faceted navigation, adding in some innovations of their own (see Figure 2). To facilitate more multi-select options, the interface has a clever blending of checkboxes and hyperlinks (but unfortunately does not support hierarchical facets). Some facet labels start out with checkboxes (such as Cities), indicating the ability to do a disjunction on the facet from the start, while others show a hyperlink (such as Distance Away), indicating that only one choice can be made at a time in the facet. After one of these choices is made, it filters the results, but is not added to the query explicitly; rather, the other choices continue to be shown as hyperlinks with the currently selected choice shown in bold. This is a departure from the standard approach in which selecting a label removes the other choices for that label.

On the downside, additional categories are tucked away under Features, which suggests that the additional ones will rarely be seen or used. This view also does not show previews of number of hits; it is potentially confusing to do so when disjunctions are allowed; this is a tradeoff in the interface design that must be weighed.

But the innovation of interest here is that Yelp modifies the use of keyword search, using the terms typed in to change the order of labels shown within facets. For example, searching for "restaurants" within the area of "kirkland, wa" returns facets labeled Sort By (best match or best reviews), Cities, Distance, Features, Price, and Category. In the case of the screenshot, the latter is type of restaurant; initially the first few types of restaurant shown are Chinese, Indian/Pakistani, Japanese, and Sushi Bars, with a link to show more. However, if instead the initial query is "italian restaurants" the labels shown under Category are Italian, Restaurants, Pizza, and Mexican. If the query is changed to "italian restaurants", the choices shown are Dim Sum, Chinese, Restaurants, Bakeries, Asian Fusion, and other Asian

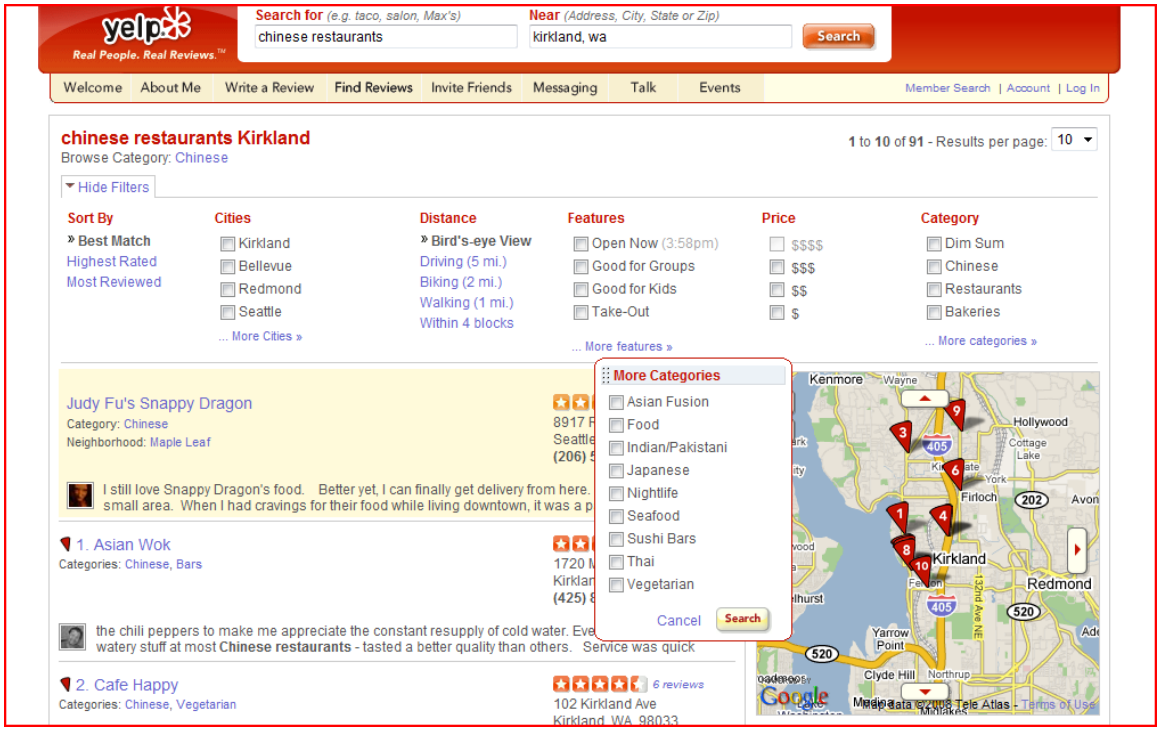


Figure 2: Yelp's new faceted search interface, modeled after that of eBay Express, but with some innovations (see text).

food categories.

Thus, this interface modifies the labels shown beneath the facets to match similar but expanded concepts related to the keyword query. It does not move out of the Restaurants domain into other topics such as Shopping, which would not be appropriate. But a query on "Asian" alone changes the Category facet to show choices such as Grocery alongside restaurant types such as Asian Fusion.

How does this behavior differ from standard (Flamenco-style) faceted navigation when given a keyword query? In Flamenco, the items that match the query determine which facet labels are shown. So a query on "chinese" would return all documents that contain that word or are assigned that label, and would show the aggregation of facet labels that are assigned to those retrieved items. These may well include Grocery and Dim Sum. But Yelp appears to be doing something more calculated. For example, a query on "dim sum" shows the categories Dim Sum, Chinese, Seafood, Food, and Restaurants, but the hits returned contain other categories including Grocery and Korean.

This interface also eliminates entire facets when not applicable to the chosen category. Chosing Beauty & Spas eliminates the Meals Served facet and brings up the By Appointment Only facet, which is not shown for Groceries. However, the mechanism does not work perfectly. For example, Beauty Salon & Spa also brings up Nightlife, Nurseries & Gardening, and Wineries. Selecting Beauty and Spa along with Wineries and Takes Credit Cards brings up an interesting collection.

5. FACETS ON MOBILE INTERFACES

Can faceted navigation be moved to the small interfaces of mobile devices? The Fathumb project at Microsoft Research [5] attempts to do just that, with a clever restriction on the number of facets, using positioning to mirror that of the number pad of a typical cell phone (see Figure 3). The results are promising, although hampered by the fact that the interface lends itself better to a touch screen than the indirection of clicking on the keyboard. The design also incorporates a subtle visualization to help indicate where in the navigation the user is, but as is often the case with such things, the participants in the lab study did not notice the visualization, or if they did, did not understand it (personal communication, Amy Karlson). This might change with further exposure to the design.

6. VISUALIZATIONS OF FACETED NAVIGATION

There have been a number of fascinating visualizations of faceted navigation, including a whimsical one from the WeFeelFine project (see Figure 4) and the FacetMap project from Microsoft Research [6]. These are visually engaging but take up a lot of screen space, so it is unclear what their ultimate uptake will be.

7. EXTENDING THE FACETED MODEL

Faceted navigation allows for flexible moves within a collection, but could be limiting for more ambitious information discovery tasks. In what ways can the model be extended but still retain the understandability needed by non-expert searchers? A full-fledged knowledge representation is too complex, but a representation that conservatively extends the design might be useful.

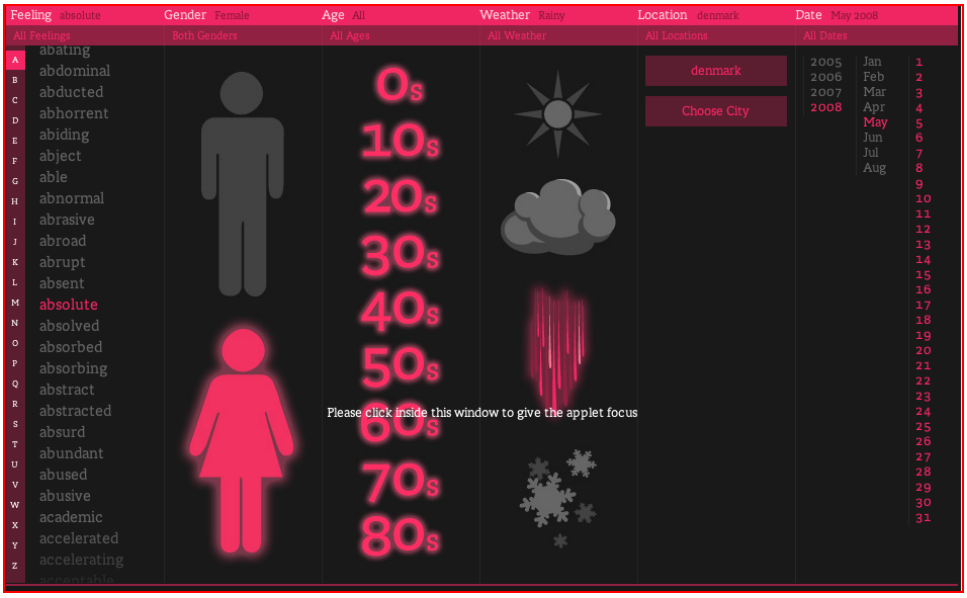


Figure 4: The whimsical faceted attribute selector from wefeelfine.org

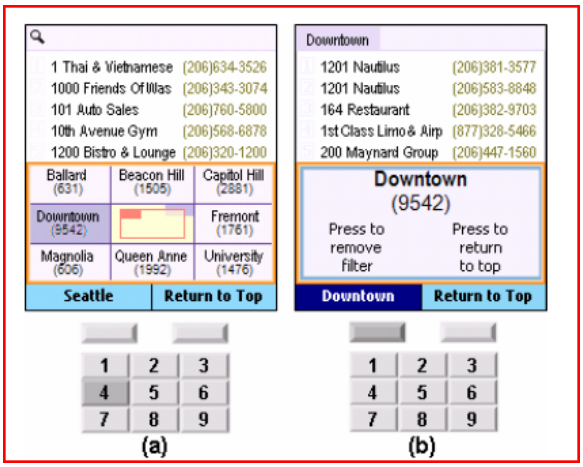


Figure 3: The FaThumb faceted interface [5] for mobile devices.

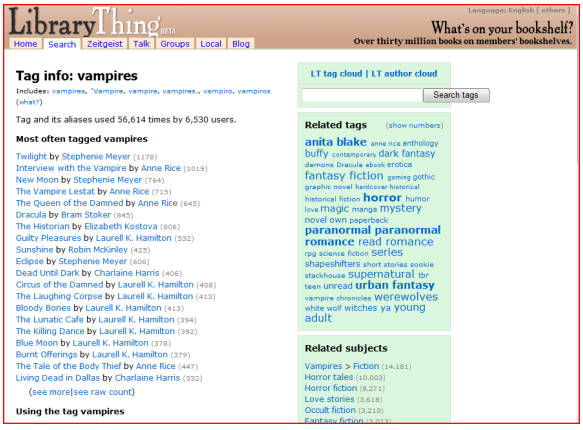


Figure 5: Tags, organized associationally and by "subject" at LibraryThing.com.

Some websites offer alternative exploration systems alongside the faceted one; Getty Images is experimenting with a design they call Catalyst. In this approach, entering one search term brings up a list of related terms in an alphabetical tag cloud. The user is asked to drag interesting terms into a selection box. The LibraryThing website shows associated terms derived from user-supplied tags, or social annotations (see Figure 6). This kind of approach might be useful for giving inspiration or brainstorming ideas, but the categories are conceptually and visually disorganized and so most likely can be improved upon.

An example of such an improvement can be seen in the work of Zelevinsky et al. of Endeca [9], who describe a promising alternative method for selecting which subject-related terms to show, in a flat list, alongside search results and conventional facets (see Figure 5). This might be further improved by adding hierarchy to the subject labels and showing more of them. It would be interesting to compare showing only the most descriptive subject terms that match the query, and then letting users navigate into relevant sub-hierarchical facets corresponding to such terms, to the standard approach of showing all the facets initially.

More recently, Huynh introduced the Parallax navigation interface over metaweb/freebase data which attempts to allow navigation of this structured data along facets as well as additional dimensions within different concept groups (see Figure 7, from http://mqlx.com/ david/parallax/). Although a promising start, it is hard to see which combinations will yield results, and seems somewhat limited by a sparsity in the number of connections allowable. But it does seem like a good start in this general direction.

8. CONCLUSIONS

Designers continue to innovate and improve the faceted navigation paradigm. However, the large-subject-space problem continues to be a tough nut to crack. Acquisition of faceted subject metadata is also a problem, although social tagging shows promise as a means towards building such

Figure 6: Using an algorithm to select relevant subject keywords, based on author keywords, for a digital library, from [9].

Figure 7: The Parallax interface extending the faceted model to related links, using structured Freebase data from MetaWeb.

structure.

Mobile computing continues to grow in popularity, and it is still an open question if faceted navigation is well-suited for the small screen. A modified variant as seen in the Fathumb project provides an encouraging direction to follow.

Information visualization is becoming increasingly prevalent for understanding and explaining information. Faceted navigation can be made more visually appealing with enhanced graphical displays, but to date it is not clear that these views enhance usability or substantially increase the number of categories that can be easily navigated.

Finally, the time has arrived to find innovative but understandable ways to extend the faceted model while at the same time retaining its essential usability. Different designers are experimenting with this but no clear good idea has emerged yet.

9. REFERENCES

[1] H. Bast and I. Weber. Type less, find more: fast autocompletion search with a succinct index. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 364–371. ACM New York, NY, USA, 2006.

[2] H. Bast and I. Weber. When You're Lost for Words: Faceted Search with Autocompletion. *ACM SIGIR Workshop on Faceted Search*, 2006.

[3] M. Hearst. Design Recommendations for Hierarchical Faceted Search Interfaces. *ACM SIGIR Workshop on Faceted Search*, 2006.

[4] M. A. Hearst, J. English, R. Sinha, K. Swearingen, and K.-P. Yee. Finding the flow in web site search. *Communications of the ACM*, 45(9), September 2002.

[5] A. Karlson, G. Robertson, D. Robbins, M. Czerwinski, and G. Smith. FaThumb: a facet-based interface for mobile search. *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 711–720, 2006.

[6] G. Smith, M. Czerwinski, B. Meyers, D. Robbins, G. Robertson, and D. Tan. FacetMap: A Scalable Search and Browse Visualization. *IEEE Transactions On Visualization And Computer Graphics*, pages 797–804, 2006.

[7] R. White and G. Marchionini. Examining the effectiveness of real-time query expansion. *Information Processing and Management*, 43(3):685–704, 2007.

[8] K. Yee, K. Swearingen, K. Li, and M. Hearst. Faceted metadata for image search and browsing. *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 401–408, 2003.

[9] V. Zelevinsky, J. Wang, and D. Tunkelang. Supporting Exploratory Search for the ACM Digital Library. *Workshop on Human-Computer Interaction and Information Retrieval (HCIR'08)*, October 2008.

Creating Exploratory Tasks for a Faceted Search Interface

Bill Kules
The Catholic University of America
School of Library and Information Science
Washington, DC
kules@cua.edu

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

ABSTRACT
In this paper we describe a process for creating and evaluating exploratory tasks for a faceted search interface. We used the tasks in an eye tracking study of a faceted library catalog search interface. We report on user perceptions of the tasks. The method is intended to be extensible to generate exploratory tasks for other types of interfaces and domains.

INTRODUCTION
Designing exploratory search tasks is an important, but challenging, requirement for successfully evaluating exploratory search interfaces. When creating any type of search task, there is a challenge of creating a realistic, representative task. When creating exploratory search tasks there is an additional burden of actually inducing an **exploratory** search. This high level goal of doing an exploratory search drives how users interpret the tasks, their relevance, and the results (Kules & Shneiderman 2008).

We set out to explore interfaces for exploratory search in a library Online Public Access Catalog (OPAC). Specifically, we were interested in studying facet use in exploratory search in a faceted-OPAC system such as the one currently in use at North Carolina State University (http://www.lib.ncsu.edu/catalog/).

Creating well-grounded, realistic exploratory search tasks was one of the primary challenges of the study design. Exploratory search tasks in a library catalog are a form of what librarians call "subject searches". We differentiate exploratory tasks because a subject search in a catalog can take place at any stage of the search process, whereas exploratory search describes the high level goal of the task. In this work, we explicitly situate the subject search at the early stage of the overall search and design tasks that induce subject search driven by a high level scenario. To create such tasks, we first needed to operationalize exploratory search for this study. Second, we needed to construct a concrete set of tasks that were appropriate for the system being used.

Operationalizing exploratory search
Exploratory tasks inherently have uncertainty, ambiguity and discovery as common aspects (White, Kules, et al.; Marchionini 2006). The searcher may not know the domain well and the information need may be ambiguous or imprecise. In addition, exploratory search typically requires retrieving multiple results to achieve the objective. This suggests several operational characteristics for exploratory search tasks:

- Answers are not found on the first interaction
- Searchers interact with the results and/or reformulate their queries
- Searchers search for multiple items

We used these characteristics to drive the development of our search tasks based on topics mined from actual usage logs of the North Carolina State University (NCSU) OPAC.

Desirable characteristics of exploratory tasks
The literature suggests a number of desirable characteristics for exploratory search. Marchionini (2006) lists exploratory tasks, characterizing them as either learning-oriented or investigative. This suggests that the high-level scenario should be described so that it involves learning or investigation. Kuhlthau (1991) describes six stages of search and predicts various types of searcher interaction. Early stages are characterized by uncertainty.

Task complexity refers to the degree of predeterminability of task performance (Byström and Järvelin, 1995). Some tasks are well established and understood (known), while others are more unique and less understood (genuine decision tasks). Problem structure, task complexity and prior knowledge have an interconnecting impact when searching. "The more complex the task, the more ill-structured it is, and the less prior knowledge the actor has." (Vakkari 1999).

Borlund (2000) advises that simulated situations include: "i. A situation which the test persons can relate to and in which they can identify themselves; ii. A situation that the test persons find topically interesting, and; iii. A situation that provides enough imaginative context in order for the test persons to be able to relate and apply the situation."

Kules & Shneiderman (2003) used four simulated work tasks for journalists constructed around an exploratory search task to evaluate a faceted web search interface, drawing on Yee et al. (2003), which included open-ended tasks that were constructed with similar objectives.

This brief review suggests that exploratory search tasks should:

- Indicate uncertainty, ambiguity in information need and/or need for discovery.

- Suggest a knowledge acquisition, comparison, or discovery task
- Be an unfamiliar domain for the searcher
- Provide a low level of specificity about:
  - The information necessary for their search
  - How to find the required information
  - How to recognize the required information
- Be a situation which the test persons can relate to and in which they can identify themselves
- Be a situation that the test persons find topically interesting
- Be a situation that provides enough imaginative context in order for the test persons to be able to relate and apply the situation.

Not all of these are practical or feasible, however. For example, in our study, we constrained the searchers to use the faceted OPAC, which indicates a very specific direction for "how to find the required information." Also, we may not be able to control for prior knowledge. Instead it may be more practical to measure it and analyze that factor.

TASK CONSTRUCTION

We followed a two-step approach to create the exploratory search tasks used in our study. First, we mined log files from the NCSU OPAC for topics that met a series of criterion. Second, we plugged the topics extracted from the log analysis into "task templates" that we designed to motivate an exploratory search. Each of these steps is described in more detail in the sections below.

Topic extraction from log data

We had the benefit of partnering with NCSU on this study and thus had access to several days of anonymized log data from their OPAC. The log files provided a list of queries issued to the OPAC. For each query, the keyword string and list of facets selected was recorded in the log. While we could not determine a searcher's exact intent from the data, we looked for searches that had characteristics of exploratory searches based on the operationalized characteristics outlined in the first section of this paper. Additionally, we were interested in facet usage, so we included it as a criterion when examining the log data. We scanned the log files looking for searches in which:

- *Answers were not found on the first interaction* – we looked for searches with multiple page views.
- *Searchers used facets* – we looked specifically for the use of one or more of three facets: Subject, Region, and Time Period. We selected these three because they were commonly used and they were not specific to NCSU's library system.
- *Searchers interacted with the results and/or reformulate their queries* – we looked for searches in which facets were added to an original query.
- *Searchers searched for multiple items* – again we looked for searches with multiple page views.

These criteria indicate that the user did not find the results on their first interaction and either reformulated the search or interacted with the results. We disregarded instances where a facet chosen was either identical or similar to the search terms (for example: a search for "cotton management" modified by selecting the Subject facet "cotton"). We also disregarded instances where the user needed to use a "show more" option to see additional facet values because we wanted to focus on facets visible from the initial results page. For example, from one log file, we observed queries for the search term "British History" with the facets "History" (subject) and "Twentieth Century" (time period). From these log entries, we developed a candidate topic "British History". We intentionally included facets in the task creation process, because our goals were to study searcher behavior in this context.

Mining the log data for searches that involved multiple interactions could lead to searches that were problematic rather than exploratory. For example, a bad interface, or poor match between facets and the task could lead to multiple interactions. The refinement step described below should help address such tasks.

Plugging the topic into a task template

To help achieve the goal that the exploratory search tasks motivate consideration of multiple items, we developed a task template that involved finding multiple items – which the specific candidate topics could be plugged into. The objective of the template was to situate the participant in a familiar situation in which multiple items would need to be found. Since we recruited participants from a university population, we used a task that involved writing a paper for a class..The basic form of the template is shown below:

> *Imagine that you are taking a class called _____. For this class, you need to write a paper on the topic _____. Use the catalog to find two possible topics for your paper. Find three books for each topic.*

Based on prior experience creating exploratory search tasks, we asked participants to find specific target numbers of topics and books.

Task refinement

Once candidate tasks were created, we refined them by conducting a set of searches related to the topics on the NCSU OPAC. The purposes of this step were to: 1) clarify the wording of the task, 2) insure that the task was not too easy to qualify for use in an exploratory search, and 3) make sure that the task benefited from using facets (since facet use was a focus of our study). To do this, refined the tasks such that:

- Facet values matched one or more terms in the task; either exactly or a semantically close term
- The first 10 results did not answer the task. If the task was too easy, it would not require exploratory search.

- The facets were useful without having to click the "show more" link for the facet.

Using the example started in the previous section, we found that the query "British History" resulted in many relevant results in the top ten results returned. We then explored other topics that could be added to make the topic more challenging. By looking at the facets presented in the OPAC, we found that by adding the topic of "Colonies", the task met our criterion. Thus, the final topic was "the relationship between Great Britain and its Colonies in the Twentieth Century". The tasks generated by the process were reviewed by library science experts, and then pilot tested and further refined with three participants.

Resulting exploratory search tasks

Using the process described in the preceding sections, we developed four exploratory tasks (see A–D below). We also used two known-item tasks (E and F) based on a previous NCSU study. This was to permit comparisons with that study. The final tasks used in the study are given below.

A. Imagine you are taking a class called "Feminism in the United States". For this class you need to write a research paper on some aspect of the U.S. feminist movement, but have yet to decide on a topic. Use the catalog to find two possible topics for your paper. Then use the catalog to find three books for each topic so that you might make a decision as to which topic to write about.

B. Your professor wants you to write a paper comparing the textile industry in three countries in three different continents. Use the catalog to find three countries which have a textile industry about which books have been written. Find three books for each country.

C. Imagine you are taking a class titled "Great Britain and its Colonies in the Twentieth Century". For this class you need to write a research paper on some aspect of the relationship between Great Britain and its Colonies in the Twentieth Century but you have yet to decide on one. Use the catalog to find two possible topics for your paper. Then use the catalog to find three books for each topic so that you might make a decision as to which topic to write about.

D. You are taking a class called "History of the Olympic Games" for which you need to write a research paper. You have yet to decide on a specific topic for this paper. Use the library catalog to explore possible topics and find two. Then find at least three books for each so that you might make a decision as to which topic to write about.

E. Your professor has suggested that your group begin your project on Conservation and Biological Diversity by looking up background information in a book titled Firefly encyclopedia of trees.

F. You are working your way through the Harry Potter books and are ready to read the next one on your list, titled "Harry Potter and the Goblet of Fire".

METHODS

Our broader goals in this research were to investigate facet use in exploratory search when using a library OPAC. Generating a set of well-grounded, representative tasks that would induce exploratory search was a significant challenge in the study design. As part of the study, we included metrics and measures to give us feedback on the tasks to see if we had achieved our goals for task creation. In this section, we present details of the study as they relate to evaluating the tasks.

Twenty-one participants were recruited from the University of Maryland at College Park (UMD) to participate in this study. Of these, data was successfully collected from 18 (two sessions were unsuccessful due to system problems and we were unable to calibrate the eye tracker for one participant). The testing system was a web-based, faceted OPAC interface based on a modified version of the North Carolina State University library catalog of over 1.8 million titles. The study was conducted in the Human-Computer Interaction Lab at UMD using a computer equipped with an eye-tracker. Results related to the eye-tracker are outside the scope of this paper and will be reported elsewhere. Data was collected about the searches issued, the results selected, and the facets used for each task.

The participants were shown a 90 second video demonstration of the interface. They then conducted six short searches motivated by the tasks, completed a questionnaire and provided a retrospective verbal report while viewing screen video of their searches with their gaze pattern overlaid. The exploratory tasks were presented first, followed by the known item tasks. Within each task type, presentation order was counterbalanced to minimize order and learning effects. In between each task, participants completed a questionnaire with five questions about their experience. All responses were given as ratings on 5-point Likert-type scales (anchors shown in parenthesis):

1. How familiar were you with this subject when you began this task? (1 = not familiar at all, 5 = very familiar)
2. How difficult was it to accomplish this task? (1 = very difficult, 5 = very easy)
3. I am confident that I fulfilled the task asked of me. (1 = strongly disagree, 5 = strongly agree)
4. To what extent did completing this task involve finding a single item versus finding multiple items? (1 = single item, 5 = multiple items)
5. To what extent did you change what you were looking for based on the results you found? (1=not at all, 5=a lot)

Additionally, at the end of the session, we asked users to perform a card sort to group the six tasks according to what tasks they thought were most similar.

RESULTS

For the exploratory searches, none of the participants found their answer(s) on their first interaction – they all interacted with multiple pages.

Perceptions of tasks

Table 1 shows the averages and standard deviations (in parenthesis) of the participants' perceptions of the

exploratory and known item tasks based on the five questions asked after each task. Participants were slightly more familiar with the known item tasks and found them somewhat easier. They were also slightly more confident that they had accomplished the indicated task. Participants clearly differentiated between the number of items that each task required (single vs. multiple). They also changed what they were looking for more for the exploratory tasks.

	Exploratory n=72 avg (stdev)	Known-item n=36 avg (stdev)
1. Familiarity	2.6 (1.39)	3.0 (1.80)
2. Difficulty **	4.0 (0.91)	4.9 (0.23)
3. Confidence **	4.2 (0.94)	4.8 (0.80)
4. Single/Multiple **	4.2 (0.92)	1.4 (1.15)
5. Changed goal **	3.3 (1.33)	1.1 (0.40)

** significant difference found between exploratory and know-item at $p<0.001$ using two-tailed T-test with $\alpha = 0.05$

Table 1. Overall Perception Ratings

Card sorting the tasks

We wished to learn whether participants perceived the exploratory tasks as similar to each other and different from the known item tasks, so we asked them to group the tasks "and put the ones that are the most alike together into groups." Of the 17 participants who completed this step, all 17 put the two known item tasks (E & F) in their own group. Nine of the participants grouped tasks A, C, and D together, placing B separately. Three put A, B, C, D all together. The remainder had various grouping of A, B, C, D. When asked about task B, the explanations focused on the geographic nature of the task and the fact that it asks for books instead of topics, as the other three do. We anticipated the strong distinction between exploratory and known-item, but the sub-distinction of tasks within the exploratory set was unexpected and suggests that participants considered the geographic/topical and books/ideas differences to be important aspects of the nature of the tasks.

Limitations

Our operational definition of exploratory search was fairly narrowly tailored to the goals of this study. Future work should incorporate additional dimensions. Task complexity, in particular, is an important dimension – multiple levels of complexity in the task descriptions could be evaluated to determine what levels of complexity induce exploratory search behavior. Only one high level scenario was used for the task template. A broader range of scenarios should be explored and tailored to more directly fit test participants, consistent with Borlund's (2000) recommendations for simulated work tasks.

DISCUSSION AND CONCLUSION

Overall, the tasks achieved our objectives. Based on the participants' perceptions of the tasks, we believe that our procedure for task generation led to well-grounded, realistic tasks that did elicit exploratory search behavior for the exploratory tasks. The exploratory tasks met the desired characteristics we outlined as goals: relatively low initial topic familiarity, require multiple items to be considered, and some ambiguity as to the final answers (as indicated by the confidence and changed goal measures). The difference in task B suggests that searchers differentiate between the indicated object (books vs paper topics) and by the nature of the facets (topical vs geographic).

This paper suggests a principled way of task building that incorporates consideration of the dimensions of the task, then building and refinding the task description while taking into account both the broader dimensions of exploratory search and the pragmatics of the particular search system and collection technique. We hope that this task development strategy is a first step toward making tasks more comparable across studies.

ACKNOWLEDGEMENTS

This research was supported by a grant from Catholic University and in part by a grant from the NSF/Library of Congress (IIS 0455970). Thanks to Matt Banta for extensive assistance in implementing the study. We thank Tito Sierra, Jason Casden, and Joe Ryan at NCSU for development of the faceted OPAC interface, log analysis and contributing to the task construction. Thanks also to Doug Oard and members of the UMD HCIL for the use of their facilities and eye-tracker.

REFERENCES

Borlund, P. (2000). Experimental components for the evaluation of interactive information retrieval systems. *Journal of Documentation*, **56**(1), 71-90.

Byström, K. & Järvelin, K. (1995). Task complexity affects information seeking and use. *Information Processing & Management*, **31**(2), 191 - 213.

Kuhlthau, C.C. (1991). Inside the search process: information seeking from the user's perspective. *Journal of the American Society for Information Science*, **42**(5): 361-371.

Kules, B. and Shneiderman, B. (2008). Users can change their web search tactics: Design guidelines for categorized overviews. *Information Processing & Management*. 44(2): 463-484.

White, W. Kules, B., Drucker, S., schraefel, m.c. (2006). Introduction. *Commun. ACM* 49(4): 36-39.

Vakkari, P. (1999). Task complexity, problem structure and information actions. Integrating studies on information seeking and retrieval. *Information Processing & Management*, **35**(6), 819-8

Yee, K.-P., Swearingen, K., Li, K., & Hearst, M. (2003). Faceted metadata for image search and browsing. In *Proc. SIGCHI*, Ft. Lauderdale, FL (pp. 401-408). New York: ACM Press.

Personal Information Organization and Retrieval Using an Activity-Based Desktop Interface

Stephen Voida
Department of Computer Science
University of Calgary
2500 University Drive NW
Calgary, AB, Canada T2N 1N4
svoida@ucalgary.ca

ABSTRACT

The venerable desktop metaphor is beginning to show signs of strain in supporting modern knowledge work. In this position paper, I examine how the desktop metaphor can be re-framed, shifting the focus away from a low-level (and increasingly obsolete) focus on documents and applications to an interface based upon the creation of and interaction with manually declared, semantically meaningful activities. In this position paper, I present the information organization and retrieval aspects of the Giornata desktop interface in detail and describe how I implemented the system to support a longitudinal deployment. I conclude with a sampling of the findings from the user study and propose opportunities for future work based on the experience.

Categories and Subject Descriptors

H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—Graphical user interfaces, H.3.2 [**Information Storage and Retrieval**]: Information Storage—File organization, H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—Information filtering.

General Terms

Design, Human Factors.

Keywords

Activity-based computing, desktop computing, context-aware computing, knowledge work, Giornata

1. INTRODUCTION

The desktop metaphor was developed over 30 years ago at Xerox PARC. The interaction techniques comprising the desktop interface responded to the needs of knowledge workers and the capabilities of computer technology in that era. The presence of a desktop "surface" behind application windows provided spatially oriented, persistent storage for icons representing files, application shortcuts, disk drives, and, eventually, the computer, itself.

New models for information storage have begun to disrupt the original model derived from information management on the physical desktop, which maps individual documents to individual files in the filesystem and each of these documents to a single window. Piles [10] and BumpTop [1] investigated grouping behaviors similar to those provided for windows via virtual desktops, but did so at the level of managing iconic representations of documents and applications where they are stored. Some information types—most prominently, e-mail, but also media files such as music and photos—are often not managed through the traditional desktop interface but are instead managed in separate information "silos" [2], stored separately from "traditional" documents and accessible only through a dedicated application, such as an e-mail client or a music "jukebox" application. The migration to more web-based storage and manipulation of documents is extending this distance between the desktop metaphor and individual documents; it is not uncommon to have a window be the *only* representation of a document locally, with the file itself stored in a web-based repository.

The Giornata[1] prototype system demonstrates how the traditional desktop metaphor can be re-framed to retain the spirit of simplified interaction with applications and files and yet better support contemporary knowledge workers' practices by emphasizing *activity* as the primary organizing principle in the interface[2]. Although other research systems have proposed using tasks or activities to organize personal information (e.g., [5, 6, 11]), Giornata is unique in that it attempts to closely integrate the activity-based tools directly into the desktop interface, providing semantically meaningful resource organization and retrieval capabilities without displacing the work practices already commonly used by knowledge workers. Giornata's enhanced desktop serves not only as a display space for application windows, but also serves as an active folder for documents and other information items associated with the current activity (Figure 1). Giornata utilizes lightweight activity- and document-tagging capabilities that enable informal and evolutionary resource organization, as well as integrating seamlessly with the search functionality provided by the operating system.

In this position paper, I present the information organization and retrieval aspects of the Giornata desktop interface in detail and describe how I implemented the system to support a longitudinal deployment. I conclude with a sampling of the findings from the user study and propose opportunities for future work based on the experience.

2. ACTIVITY-BASED INFORMATION ORGANIZATION IN GIORNATA

Giornata takes as its starting point the virtual desktop metaphor of the Rooms and Kimura systems [4, 8]. In addition to providing straightforward activity "spaces" into which focused work on single activities can be concentrated and their constituent

Copyright is held by the author/owner.
This position paper was presented at the Second Workshop on Human-Computer Interaction and Information Retrieval (HCIR 2008), October 23, 2008, Redmond, WA, USA.

[1] *Giornata* is Italian for "day's work," and, in the context of *buon fresco* (wet plaster) painting, denotes the area of a painting—the amount of work—that can be completed in a single session.

[2] This paper represents a subset of a larger research agenda in developing activity-based desktop systems grounded in cognitive theory and observations of real-world practice. An extended version of this paper has been previously published elsewhere [14, 15].

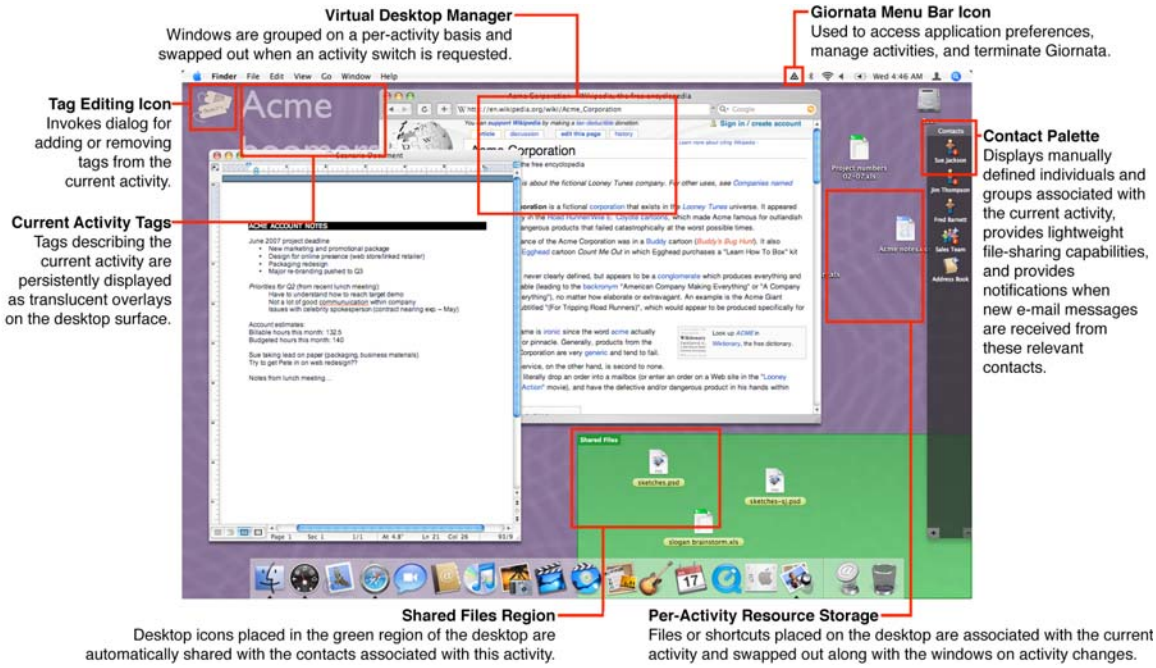


Figure 1. The Giornata interface. In this screenshot, an individual is engaged in managing a particular client's business account. There are several tags (including the client's name, "Acme"), two open windows, six files (three of them shared), three colleagues, and one group associated with this activity.

components organized, Giornata provides a number of novel information organization and collaboration features.

In Giornata, each activity is associated with a corresponding virtual desktop. In order to support fluid—and often fast-paced work, the system enables creation of a new, empty, untagged activity using a single keystroke. This action hides all on-screen windows and desktop contents, presenting a clean canvas on which work can begin on a new activity without distraction or the need to manually manage digital clutter.

Giornata allows an individual to navigate among open activities using a status bar menu, accelerator keys, or a quick activity switcher, which operates using the same interface principle as the application switching service available both in Windows (invoked using alt + tab) and OS X (via command + tab).

2.1 Activity-Based Resource Storage

In Giornata, the desktop serves not only as a display space for application windows, but also as a readily accessible folder for documents and shortcuts associated with the current activity. Any file saved or copied to the desktop is automatically associated with the current activity; as an individual switches among ongoing activities, these resources are "swapped out" along with application windows and temporarily stored in a folder associated with the activity until the activity is resumed. The effect of this feature is that the desktop workspace is automatically repopulated with the files, folders, and other information resources associated with each activity as an individual's focus changes. This behavior is similar to the approaches taken by Time-Machine Computing [13] and the Context Browser [12], with the main difference being the underlying organizing principle determining the visibility of the desktop's contents, Giornata's being activity instead of time.

These capabilities filter the information displayed on the screen at any time to the most relevant applications, information resources, contacts, and communications. The act of retrieving information related to an ongoing activity is reduced to switching to that activity (if necessary), revealing the contents of the desktop using OS X's Exposé interaction technique, and performing a visual search of the items on the desktop surface. The emphasis on locating items of interest within an activity takes advantage of individuals' natural inclination to associate information resources with their context of use, as well as the strong spatial organization practices observed by Kidd and Malone in their studies of knowledge work practice [7, 9].

2.2 Activity Tagging

Each activity in Giornata can be annotated with optional, freeform tags to describe its semantics. Activities are initially created without tags; the ability to create and work in an unnamed desktop allows work to proceed even when an individual might not know the significance or eventual meaning of an activity at its outset.

An activity's tags help individuals identify the activity in which they are currently working and distinguish among background activities. The active activity's tags are persistently visible, rendered over the desktop wallpaper; they can also optionally be displayed in the menu bar.

When an activity has one or more tags associated with it, these tags are transferred to each file touched over the course of working in that activity[3]. This design serves to "stamp" files with information about the context in which they were created or edited, and helps to overcome the burdensome process of manually adding semantic metadata to each file associated with an activity, an approach similar to that taken by Dourish et al. [3]. It also allows documents that are shared across multiple activities to be stored elsewhere in the filesystem and still "inherit" tags from

[3] File tagging can optionally be extended to include e-mail messages, iCal calendar entries, and Address Book cards, as each of these objects are represented by individual files in OS X.

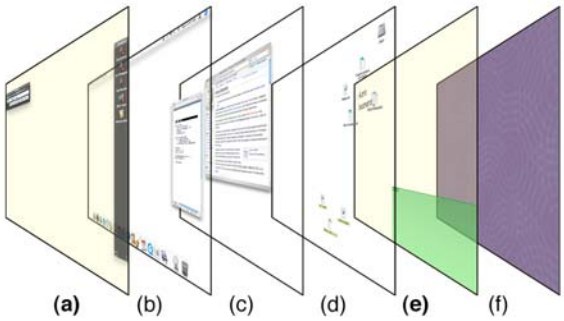


Figure 2. Explicit and implicit interaction layers in the Giornata system, and their relationship to existing window manager interaction layers. This figure illustrates the interaction layers of Figure 1: (a) Giornata's explicit interaction layer; (b) the system menu and Dock; (c) application windows; (d) desktop icons; (e) Giornata's implicit interaction layer, including activity tag display and sharing space; and (f) the desktop wallpaper.

all activities in which they are used. Because the Spotlight framework automatically indexes these tags, individuals can find information resources using the semantically meaningful tags they assigned to the activity, regardless of where the files associated with the activity are actually stored on the disk.

As an individual comes to understand the meaning of a particular activity, she can edit the activity's tags by clicking on a tag icon on the desktop surface. She is then given the option to tag the activity's files from that point forward or to retroactively tag all of the files previously associated with the activity as well. This ability to create post hoc tags on activities and files enables individuals to refine the meaning of an activity as that meaning emerges or changes over the course of the work. It also helps to ensure that the system's activity representations are sufficiently flexible to adapt to the individual's evolving work environment.

One of the fundamental design goals of the Giornata system is to reduce the interaction costs of associating semantically meaningful metadata with individual digital artifacts. Although an information retrieval perspective did not explicitly inform the design of Giornata, the system provides a fundamentally different structure within which personal information is stored and retrieved on a day-to-day basis. Giornata shifts the information retrieval focus from browsing semantically impoverished file hierarchies and searching with content-based metadata to browsing by activity and searching with semantically meaningful tags. Prior empirical research suggests that this change will have a significant and positive impact on individuals' ability to find and reference their digital artifacts [2].

2.3 Implicit and Explicit Interaction Design

Giornata's interface integrates closely with the existing file and window management components of Apple OS X. The OS X window manager emulates the physical manipulation of paper on a desk by compositing application windows on various layers above the desktop file icons and wallpaper, but below system-wide interaction widgets like the menu bar and Dock (Figure 2). Giornata augments this visual stack by inserting two additional layers: an explicit interaction layer on top of all other layers (Figure 2a), providing persistent visibility of the Contact Palette and allowing individuals to control the activity management system, and an implicit interaction layer below the desktop file icons but above the background wallpaper (Figure 2e). This non-interactive layer serves as a persistent information display for information such as the current activity tags. It also passively monitors interactions with existing desktop objects (such as desktop file icons), providing the system with input as a side effect of other, typical desktop interactions.

The implicit interaction layer is a particularly powerful component of the Giornata interface design. Because it serves as a persistent information display and is "anchored" to the desktop wallpaper and rendered translucently, a quick overview of the activity state can be quickly surmised by invoking the "show desktop" feature of Exposé. The seamless augmentation of the desktop background also helps to convey Giornata's status as an integral part of the desktop environment. It also serves to reduce visual clutter, as Giornata's interface elements are typically hidden behind application windows until needed.

3. SYSTEM ARCHITECTURE

Giornata is implemented on OS X as a hybrid Carbon-, Cocoa-, and AppleScript-based application. The application is designed to run continually while an individual is logged in and provide activity-management services alongside other system applications.

I chose OS X as the host platform for the Giornata prototype for three main reasons. First, the OS X window manager already provides a framework (albeit undocumented) for creating and managing virtual desktops. Second, Apple's use of a metadata-based filesystem (HFS+), along with the tight integration of the Spotlight search engine into the desktop interface enabled us to create a robust file- and activity-tagging infrastructure that could integrate easily into users' existing information foraging practices. Third, AppleScript, a powerful and well-established cross-application scripting language that is integrated into the OS, allowed us to quickly prototype interactions with existing applications and data sources without need for modifying other applications' source code to be explicitly "Giornata-aware."

Giornata's tag manager is implemented as an Objective-C category extending Cocoa's NSFileManager class and provides additional functions for converting between activity tags and comment strings and for setting and retrieving Spotlight Comments for specified files via AppleScript. Activity tags used to annotate a file are each prefaced with an "@" character and appended to any existing contents in the Spotlight Comments field using a space character as a tag delimiter. This encoding scheme is computationally straightforward, ensuring that the system can quickly read or write tags for a large number of files without incurring significant overhead. It also provides a human-readable representation of the tags that can be viewed or edited using the Finder or used as search keywords in Spotlight.

When Giornata starts up, it launches a file-monitoring daemon to observe filesystem changes and automatically apply tags to files that are "touched." This process, running with root-level privileges, takes advantage of the *fsevents* kernel-level filesystem monitoring facility typically used by Spotlight to detect when files are created or changed so they can be indexed for rapid search. This approach ensures that Giornata "sees" any work taking place in the filesystem and allows the system to automatically tag changed files with semantically meaningful metadata without incurring any additional interaction costs.

When the daemon detects that the desktop database file has been modified, indicating that items have been added to, removed from,

or moved to a different location on the desktop, it sends a notification to the main Giornata application that an implicit input action has taken place. When this notification is received, the main Giornata application examines each of the items on the desktop using an AppleScript to determine if their desktop positions fall within the boundaries of the sharing space. When an item is found to be within this space, Giornata turns on the item's Finder highlighting (as a confirmation that the system has recognized and begun sharing the item) and adds the file to the list of shared files for the activity.

The implicit interaction layer is also responsible for maintaining per-activity desktop file storage. When an activity switch is requested, the (X, Y) position of each file on the desktop is captured using an AppleScript and then the current contents of the desktop are moved to a storage folder associated with the activity, typically in the folder named "/Users/*username*/Activities/*activity tags*". Once the desktop has been cleared, the desktop contents of the incoming activity are restored and each item is manually re-positioned at its previous location on the desktop.

4. DEPLOYMENT AND STUDY

I deployed the Giornata prototype to five participants (two university faculty members, two graduate students, and one industrial HCI practitioner), who used the system as part of their everyday work for an average of 54 days (min = 22 days; max = 82 days). For the deployment, I instrumented Giornata to log information about all activity-based interactions. At the conclusion of the deployment, I asked participants to rate the usefulness of several aspects of the system and conducted semi-structured interviews with each of the participants to elicit specific feedback about their experiences using the software.

Participants logged substantial real-world use of the system, with an average of 7.6 open activities per participant over the course of the study (SD 3.5). Participants engaged in an average of 28.2 activity switches per day (SD 15.9).

The per-activity resource storage was frequently cited as one of the "biggest wins" in using the system. All of the participants used this feature (to varying degrees), and most commented that having a place to store files without having to negotiate the hierarchical filesystem was valuable. One participant noted that routinely saving files to the desktop "feels better than filing."

The study participants were all relatively light Spotlight search users, which produced little data of note about the usefulness of incorporating activity tags into Spotlight search queries. However, most participants noted that while tagging played only a minor role in their day-to-day system use during the deployment, the real value in tagging activities and their associated documents might not be realized until the very long term (e.g., six months or more). This suggests a potentially fruitful direction for future research: evaluating the relative use of spatial information retrieval using views filtered using semantically meaningful activity boundaries, as compared to search-based retrieval over an extremely long deployment (e.g., a year or more of continuous system use).

5. CONCLUSIONS

The Giornata system illustrates how activity-based information organization tools can be incorporated directly into the desktop interface to provide powerful, semantically meaningful storage and retrieval capabilities for knowledge workers. The Giornata software provides a platform upon which further research can be carried out exploring the ways that implicit and explicit representations of activity might affect information storage and retrieval practices in knowledge work.

6. ACKNOWLEDGEMENTS

This research was completed under the advisement of Elizabeth D. Mynatt and W. Keith Edwards as part of my dissertation research at the Georgia Institute of Technology. I would also like to thank Gregory Abowd, Blair MacIntyre, and Tom Moran for their feedback on the design of the Giornata system and the presentation of this research.

7. REFERENCES

[1] Agarwala, A. and Balakrishnan, R. Keepin' it real: Pushing the desktop metaphor with physics, piles, and the pen. In *Proc. CHI '06*, ACM Press (2006), 1283–1292.

[2] Bergman, O., Beyth-Maron, R. and Nachmias, R. The project fragmentation problem in personal information management. In *Proc. CHI '06*, ACM Press (2006), 271–274.

[3] Dourish, P., Edwards, W.K., LaMarca, A., Lamping, J., Petersen, K., Salisbury, M., Terry, D.B. and Thornton, J. Extending document management systems with user-specific active properties. *ACM Transactions on Information Systems 18*, 2 (April 2000), 140–170.

[4] Henderson, J.D.A. and Card, S.K. Rooms: The use of multiple virtual workspaces to reduce space contention in window-based graphical user interfaces. *ACM Transactions on Graphics 5*, 3 (July 1986), 211–241.

[5] Jones, W., Klasnja, P., Civan, A. and Adcock, M.L. The personal project planner: Planning to organize personal information. In *Proc. CHI '08*, ACM Press (2008), 681–684.

[6] Kaptelinin, V. UMEA: Translating interaction histories into project contexts. In *Proc. CHI '03*, ACM Press (2003), 353–360.

[7] Kidd, A. The marks are on the knowledge worker. In *Proc. CHI '94*, ACM Press (1994), 186–191.

[8] MacIntyre, B., Mynatt, E.D., Voida, S., Hansen, K.M., Tullio, J. and Corso, G.M. Support for multitasking and background awareness using interactive peripheral displays. In *Proc. UIST '01*, ACM Press (2001), 41–50.

[9] Malone, T.W. How do people organize their desks? Implications for the design of office information systems. *ACM Transactions on Office Information Systems 1*, 1 (January 1983), 99–112.

[10] Mander, R., Salomon, G. and Wong, Y.Y. A 'pile' metaphor for supporting casual organization of information. In *Proc. CHI '92*, ACM Press (1992), 627–634.

[11] Muller, M.J., Geyer, W., Brownholtz, B., Wilcox, E. and Millen, D.R. One-hundred days in an activity-centric collaboration environment based on shared objects. In *Proc. CHI '04*, ACM Press (2004), 375–382.

[12] Park, Y. and Furuta, R. Keeping narratives of a desktop to enhance continuity of on-going tasks. In *Proc. JCDL 2008*, ACM Press (2008), 393–396.

[13] Rekimoto, J. Time-machine computing: A time-centric approach for the information environment. In *Proc. UIST '99*, ACM Press (1999), 45–54.

[14] Voida, S. Exploring User Interface Challenges in Supporting Activity-Based Knowledge Work Practices. Unpublished doctoral dissertation. Georgia Institute of Technology, Atlanta, Georgia, USA, 2008.

[15] Voida, S., Mynatt, E.D. and Edwards, W.K. Re-framing the desktop interface around the activities of knowledge work. To appear in *Proc. UIST '08*. Monterey, California, USA, October 19–22, 2008.

Human-Guided Ontology Learning

Hui Yang
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
huiyang@cs.cmu.edu

Jamie Callan
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
callan@cs.cmu.edu

ABSTRACT

This paper leverages human knowledge and understanding in machine learning algorithms for constructing ontologies. Ontology construction is a highly subjective task where a human user builds a data model which represents a set of concepts within a domain and the relationships between those concepts. Personal preferences have crucial impact on manually-built ontologies, however are inadequately captured by traditional supervised machine learning approach. This paper proposes a human-guided machine learning approach, which incorporates periodical manual guidance into a supervised clustering algorithm, for the task of ontology construction. A user study demonstrates that guided machine learning is able to generate ontologies with manually-built quality and less costs. It also shows that periodical manual guidance successfully directs machine learning towards personal preferences.

INTRODUCTION

Ontology construction, or ontology learning, is an important task in Artificial Intelligence, Semantic Web and Knowledge Management. It is the process of building an ontology, a data model that represents a set of concepts within a domain and the relationships between those concepts. An ontology is about the given corpus or domain, identifies and often organizes the concepts into a tree-structured hierarchy. In most cases, ontology learning is highly subjective and task-specific. For example, when writing a literature review for human computer interaction (HCI), we may crawl the Internet for the relevant materials, sort through various documents, identify important concepts and milestones in the literature, find the important relationships between them, and organize them based on the relationships. Note that different person will have different ways to define "what is an important concept or milestone" and "what is an important relationship", and hence results in different ontologies for HCI. In general, personal preferences show crucial impact on manually-built ontologies.

In the context of ontology construction, personal preferences are represented as periodical manual guidance in guided machine learning, which combines the strengths of both human expertise and machine learning to build ontologies. In particular, human users teach the system to create a personalized, task-specific ontology by providing appropriate scaffolding, a concept in the Situated Learning Theory referring to the supports provided by a teacher to help a student achieve tasks which are not able to accomplish independently, while the system learns from such manual guidance, adjusts the learning process with appropriate changes and produces learned results by following the guidance. The teaching and learning actions occur alternatively at each learning cycle and the entire process continues until a human-satisfied ontology is built. There are two major questions for research on constructing ontologies by guided machine learning and they are:

(1) Can a guided machine learning approach produce ontologies with the same quality as manually-built ones?

(2) Can a guided machine learning approach learn from individual users and capture the distinctions among their personal preferences?

To answer the above questions, this paper studies the effects of guided machine learning on ontology construction. In particular, it employs a supervised clustering algorithm, which learns distance metrics for concept pairs in an ontology, in a guided bottom-up hierarchical clustering framework. At each human-computer interaction cycle, cluster partitions from human guidance, are taken as the training data, from which a distance metric is learned. The distance metric is then used in a flat clustering algorithm to create clusters at the higher level. A user study demonstrates that guided machine learning is able to generate ontologies with manually-built quality and manual guidance successfully directs machine learning towards personal preferences.

A GUIDED HIERARCHICAL CLUSTERING FRAMEWORK

In this section, we model the process of ontology construction as a guided machine learning framework. Given the fact that most ontologies are hierarchies in nature, we employ hierarchical clustering as the main guided learning framework, in particular, a bottom-up hierarchical clustering framework. Algorithm 1 gives the pseudo-codes for the guided hierarchical clustering algorithm. Starting from the bottom, the process builds up the ontology level by level by learning a new distance metric from the current level and applying it to the higher level. At each iteration, any flat clustering algorithm can be used to construct concept groups. The flat clustering algorithm used in this work is K-medoids [2]. We adopt Gap statistics [3] to estimate the number of clusters.

After concepts are clustered by K-medoids, if the system is in its interactive mode, it displays the learned ontology on the User Interface and waits for manual guidance. Users can interact with the system via a tool called OntoCop (**Onto**logy

Algorithm 1: Guided Hierarchical Clustering

while not satisfied or not all concepts connected in a tree
  construct groups for level i by flat clustering;
  if in interactive mode
    wait for manual guidance;
  learn distance metric function from level i;
  predict distance scores for level $i+1$;
  $i \leftarrow i+1$;
Output the tree

Construction **P**anal). Users are able to add, delete, modify concepts, drag & drop concepts around and group them accordingly. Users can also search and view the documents relevant to a concept for a better understanding of the domain knowledge when they are making decisions. When they are done with modifications to the concepts, they can upload the hierarchy to the server, which learns from the user modifications, predicts new distance scores for unorganized concepts and runs K-medoids to cluster them and returns the new hierarchy to the user.

In an uploaded hierarchy, there are many concept groups, each contains a parent concept and a group of child concepts. We call such concept groups "*ontology fragments*". From an uploaded hierarchy, which usually is a partial ontology, we decompose it into ontology fragments and use them as manual guidance in the learning process. In the proposed bottom-up approach, the grouping information in ontology fragments at the lower levels are used to estimate a distance metric function, which then predicts the distance scores for concepts at the higher levels.

INCORPORATING MANUAL GUIDANCE

In Figure 1, the ontology fragments suggest that (child, maker) is close since they are in the same group, (sport hunter, trophy hunter) is also close, (sea ice habitat, child) may be far away since they are in different groups. The goal is to find a mapping from such grouping information to their semantic distances and then use the mapping function to predict the semantic distances for ungrouped concept pairs such as (habitat, person) and (habitat, territory). The mapping is required to give reasonable scores to concept pairs such that (habitat, territory) is closer than (habitat, person).

We propose a supervised clustering algorithm based on distance metric learning [4]. In particular, the ontology construction problem is modelled such that at each time, a set of concepts $\mathbf{x}^{(i)}$ on the ith level of the ontology hierarchy is under consideration. Another training input is a distance matrix $\mathbf{y}^{(i)}$. An entry of this matrix which corresponding to concept $x_j^{(i)}$ and $x_k^{(i)}$ is $y_{jk}^{(i)} \in \{1, 0\}$, where $y_{jk}^{(i)} = 0$, if $x_j^{(i)}$ and $x_k^{(i)}$ in the same cluster; 1, otherwise. The training data consists n levels of training concepts $\mathbf{x}^{(1)}$, $\mathbf{x}^{(2)}$, ..., $\mathbf{x}^{(n)}$, each with $|\mathbf{x}^{(1)}|$, $|\mathbf{x}^{(2)}|$, ..., $|\mathbf{x}^{(n)}|$ concepts. Each set $\mathbf{x}^{(i)}$ represents a set of concepts at the level indexed by i. For each set of training data, the correct partition (clustering) are given via distance matrices $\mathbf{y}^{(1)}$, $\mathbf{y}^{(2)}$, ..., $\mathbf{y}^{(n)}$.

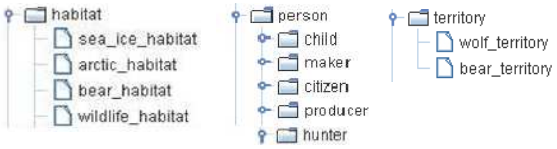


Figure 1. Ontology Fragments

In the distance matrix, within-cluster distance is defined as 0 and between-cluster distance is defined as 1. From the training distance matrix, we would like to learn a good pairwise distance metric function which best preserves the regularity in the training distance matrix. In our work, the estimated pairwise distance metric function is represented as a Mahalanobis distance [4].

$$d_{jk} = \sqrt{||x_j - x_k||^T A||x_j - x_k||}$$

Theoretically, the parameter estimation problem in our settings is to get *A* such that the expected loss is minimized. The loss function is minimized through minimizing the squared errors. The optimization function is then defined as :

$$\min_A \sum_{j=1}^{|\mathbf{x}^{(i)}|} \sum_{k=1}^{|\mathbf{x}^{(i)}|} (y_{jk}^{(i)} - \sqrt{\Phi(x_j^{(i)}, x_k^{(i)})^T A \Phi(x_j^{(i)}, x_k^{(i)})})^2$$

$$\text{subject to } A \succeq 0$$

where $\Phi(x_j^{(i)}, x_k^{(i)})$ represents a set of pairwise underlying feature functions, where each feature function is $\phi_d : (x_j^{(i)}, x_k^{(i)}) \mapsto r \in \Re$ with d=1,...,$|\Phi|$. The underlying feature functions evaluates the relationship between $(x_j^{(i)}, x_k^{(i)})$ from various aspects. The next section will give more details about the feature functions. *A* is a parameter matrix, which weighs the underlying distance feature functions.

Given the learned parameter matrix *A*, it is easy to generate distance metric for any pair of unmeasured concepts. By calculating the distance for each concept pairs, we obtain the entries in a new distance matrix $\hat{\mathbf{y}}^{(i+1)}$, which contains the distance scores for concepts at the $(i+1)^{th}$ level. Note that previously they were unmeasured and unorganized. The scores are then used to produce partitions.

In a nutshell, in the guided hierarchical clustering framework, the learner requests for manual guidance at each learning cycle, and adjusts the learning of the distance metric accordingly. In particular, by taking into account a user's modification to the ontology, the system learns from his/her personalized grouping of concepts.

FEATURES

The distance metric learning process models a distance metric as a function of some underlying feature functions, where each feature function is a measurement of how distant two concepts are. Features used in this work are a balanced mixture of *statistical, contextual* and *knowledge-based* distance functions. *Statistical Features* are basically various

forms of term (co-)occurrences in corpora, which are statistical evidence of how distant two concepts are. In particular, we use raw and log frequencies of term occurrences for a single concept, which is at the diagonal entries of a distance matrix, and raw and log frequencies of term co-occurrences for a concept pair. *Contextual Features* measure the concept similarity based on the distributional hypothesis. There are two kinds of contextual features used in this work. The first measures the number of word overlaps between the subjects/objects of verb predicates where each of the two concepts is the object/subject. For example, for concepts "polar bear" and "seal", habitat(polar bear, arctic ice) and habitat(seal, sea ice) are two corresponding verb predicates, where the two concepts are the subjects. The word overlap between the objects is 1 ("ice" in this case). The second measures the number of word overlaps between noun or adjective modifiers in front of two concepts. For example, the overlap between modifiers in "high blood pressure" and "peer pressure" is 0. *Knowledge-based Feature* used in this work is the number of word overlaps between the Web definitions of two concepts, for instance, for a concept pair (habitat, arctic sea) we issue query "define:habitat" and "define:arctic sea" to Google search engine. The Web definitions are then compared and the feature function outputs the number of word overlap after removing the stopwords. Note that Web definitions for concepts are mainly from Wordnet. All values from the above feature functions are normalized into $[0, 1]$ by dividing by the maximum possible values.

A USER STUDY AND EXPERIMENTAL RESULTS

To evaluate the system performance and answer the two questions posed at the beginning of the paper, a user study has been conducted for the task of ontology construction. The task is defined in the domain of public comments, where administrative agencies of the U.S. government seek comments from stakeholders and the public to issue draft versions of proposed regulations and respond in the final rule to substantive issues. The situation given in the evaluation is that the agencies need to organize the relevant materials into rule-specific ontologies based on the their actual needs.

We collaborated with an independent coding lab to conduct the user evaluation. Twelve professional coders familiar with the problem domain participated in the experiments. They were divided into two groups, four for the manual group and eight for the interactive group. Users in the manual group were asked to construct ontology with the concept candidates produced by the system in a bottom-up fashion until they felt satisfied with their work or reaching a 90-minute limit (which is carefully evaluated by the experiment designers). The interactive group were asked to work interactively with the system until they felt satisfied with the work or reaching a 90-minute limit. Each user in the interactive group worked on organizing the concept candidates for a few minutes, then uploaded the modified hierarchy to the system; then the system learned from user feedback, produced a new hierarchy and returned it to the user. It is a user's decision to continue modifying the ontology and teaching the system to learn or stop. Both groups used the same editing tool provided in OntoCop, such as deleting, adding a node, dragging and dropping a node, promoting a node to the higher level, undoing previous actions, etc. The set of concept candidates given to both groups were the same.

Table 1. Intercoder Agreements on Parent-Child Pairs

	manual-manual	manual-interactive	t	p
wolf	0.55	0.55	0	0.5
polar bear	0.44	0.46	0.21	0.42
mercury	0.61	0.51	1.89	0.03

There are four public comment data sets used in the experiments, namely "toxic release inventory (tri)" (Docket id: USEPA-TRI-2005-0073),"wolf" (USEPA-RIN-1018-AU53), "polar bear" (USDOI-FWS-2007-0008),"mercury"(USEPA-OAR-2002-0056). The vocabulary sizes of each dataset are 12,838, 51,938, 67,110 and 102,503, which result in 248, 795, 351, and 1084 concept candidates for each dataset respectively. Among these four datasets, "tri" is the one with the smallest vocabulary and used for tool training for both manual and interactive users. The experimental results generated on "wolf", "polar bear" and "mercury" datasets are reported in the following sections.

For a given ontology, a list of all parent-child pairs in the hierarchy are generated. Performance metrics for parent-child pairs measure whether a concept is assigned to the correct parent. In section we use the intercoder agreement as the performance metric while in section we use the F3-measure.

Quality of Constructed Ontologies

This experiment investigates whether the proposed guided machine learning approach is able to produce ontologies with the same quality as manually built ones. We compare the intercoder agreement between two manual runs and that between one manual and one interactive run in this experiment. The intercoder agreement measured by Cohen's Kappa between two manual runs is averaged over 4x3=12 pairs of manual-manual runs. The intercoder agreement between manual and interactive runs is averaged over 4x8=32 pairs of manual-interactive runs. Table 1 shows the averaged intercoder agreements and the significance test results for parent-child pairs and sibling pairs respectively. We can see that both the intercoder agreement between manually built ontologies and that between manual-interactive runs are within the range of 0.44 to 0.61, which indicates moderate agreement. We also observe that manual-interactive intercoder agreement is comparable with manual-manual intercoder agreement, which indicates that the guided machine learning approach is able to produce the same quality ontologies as humans do. A series of one-tailed t-tests also confirm it. Almost all significant test results are not significant, $t < 2$ and $p > 0.01$, which show no statistical significant differences from manually-built ontologies and interactively-built ontologies. The results demonstrate that guided machine learning is able to produce the same quality ontologies as humans do.

Costs of Constructing Ontologies

Table 2. Average Manual Editing Costs

	add	delete	move	name change	undo	total
manual	56.25	200	2806.75	70.25	19	3152.25
interactive	20.17	129	1693.17	39.5	7.83	1889.67

Table 3. Ontology Construction Duration

	wolf	polar bear	mercury	**average**
manual	1:24	1:22	1:33	**1:27**
interactive	1:06(0:33)	0:34(0:29)	1:05(0:30)	**0:55(0:31)**

This experiment investigates the construction costs of taking manual or interactive approach. We compare the construction logs for users from both manual and interactive groups. Table 2 shows the number of manual editings of building ontologies for three datasets. The editings include adding a (child or sibling) concept, moving a concept by drag & drop, deleting a concept, changing name for a concept and undoing previous actions. In total, interactive users use 40% less editing actions to produce the same quality ontologies as manual users do. A one-tailed t-test shows a significant reduction, t=10 and $p < 0.001$, of interactive runs in editing costs as compared to manual runs. It demonstrates that guided machine learning is significantly more cost effective than manual work.

We also compare the ontology construction duration. Table 3 shows the actual time needed to construct an ontology for both manual and interactive runs. It also shows the time in part spent by human users in the interactive runs in the brackets. In general, interactive runs save 30 to 60 minutes for building one ontology. Within an interactive run, a human user only needs to spend 31 minutes in average to construct an ontology, which is 64% less than 1 hour and 27 minutes in a manual run. It shows that guided machine learning greatly saves a human user's time to construct an ontology.

Learning from Personal Preferences

This experiment investigates the system's ability to learn from personal preferences from different users and eventually fulfil their personal needs. Figure 2 shows the changes of average F3-measure for parent-child pairs over six learning cycles. The x-axis are the learning cycles for each dataset. The y-axis indicates the averaged F3-measures.

Results for both interactive and manual users before and after each learning cycle are shown. For manual users, we use their partially constructed ontologies with 20%, 40%, 60%, and 80% modifications in the editing log and plot the F3-measures. Each individual's partial ontologies are compared with his/her own finalized ontology. The F3-measure is averaged over the 4 manual users. For interactive users, we take the ontologies that uploaded by them each time to the server and plot the F3-measures of each uploaded version and the learned ontology afterwards against his/her own finalized ontology. The F3-measure for the interactive group is averaged over the 8 members.

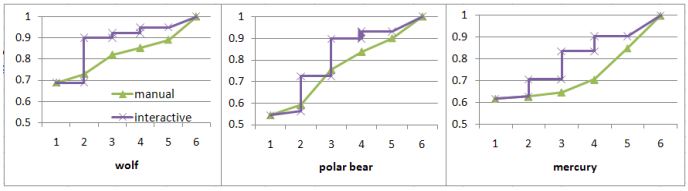


Figure 2. F3 for Parent-Child Pairs over Cycles

In Figure 2, F3-measures for both manual and interactive groups converge to 1 at the end of the learning process since it is a personalized task and each individual's finalized ontology is used as the gold standard. For interactive users, we notice an obvious performance gain between an uploaded ontology and the ontology learned automatically from it. Moreover, comparing the performances of interactive and manual users, we notice that the learning curve of the interactive users are steeper than that of the manual users. It indicates that the guided machine learning approach not only learns from personal preferences but also helps interactive users move faster towards their personal satisfaction levels.

CONCLUSIONS

This paper has shown a guided machine learning approach for the task of ontology construction. By incorporating periodical manual guidance into a distance learning algorithm in a hierarchical supervised clustering framework, it takes into account human expertise in a real-time interactive ontology construction process. A user study and experimental results demonstrate positive answers to the two questions posed on the effects of guided machine learning for ontology construction: guided machine learning is able to generate ontologies with manually-built quality and manual guidance has positive effects on directing machine learning towards personal preferences. Moreover, an analysis of the construction costs and duration shows that guided machine learning is significantly more cost effective and efficient than the manual work. Given that both guided machine learning and manual work produce ontologies with the same quality, the former becomes more attractive. Further, the results show that guided machine learning not only learns from personal preferences but also accelerates the process of ontology construction towards the personal satisfaction levels. This is very encouraging for the proposed framework.

REFERENCES

1. Z. Harris. Distributional structure. In *Word, 10(23): 146-162s*, 1954.
2. T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, 2001.
3. R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a dataset via the gap statistic. In *Tech. Rep. 208, Dept. of Statistics, Stanford University*, 2000.
4. E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. In *Advances in Neural Information Processing Systems*, 2002.

Beyond the Search Box: Helping Users Find Health Information on the Web

Kevin Duh
University of Washington
Seattle, WA
kevinduh@u.washington.edu

Shawn Medero
Healia
Bellevue, WA
shawn@healia.com

Mike Schultz, Tom Eng
Healia
Bellevue, WA
{mikes,tom}@healia.com

ABSTRACT
Internet users are increasingly relying on the Web for health information. Their information needs can often be quite complex, ranging from researching a personal illness to comparing the pros and cons of various treatments. We believe that a search interface beyond the traditional search box is necessary to support users in making informed health decisions. In this paper, we describe the search interface of Healia, a consumer health search engine, which contains advanced search features such as personalization, faceted browsing, and query suggestion. We present some analyses of the query logs to seek to understand how users interact with our search interface.

Categories and Subject Descriptors
H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces.

General Terms
Measurement, Design, Human Factors.

Keywords
Health Vertical Search, Search Interface Design, User Behavior

1. INTRODUCTION

eHealth is an industry of growing importance. The Internet provides opportunities for users to seek health advice from potentially millions of online peers and experts at any time of the day. Forrester Research found that as many as 84% of American Internet users have researched health information online in 2006 [1], and that the majority of these searches involve questions relating to specific medical conditions of the searcher or searcher's family/friends [2]. As the amount of health content proliferates on the web, there is an increasing demand for search engines and portals to organize and filter information in a personalized fashion.

Information need for health-related questions may be quite complex and varied, but we can categorize users into two general groups. In the first group, users may have been just diagnosed by a health professional with a certain illness, and is motivated to understand specific issues related to the illness in detail. Queries such as "What are the treatments for a 5 year-old with strep throat?" or "Clinical trials for diabetes in African American women" indicate the need for **highly personalized** (e.g. 5 year-old, African American women) as well as **highly specific** (e.g. treatment, clinical trials) results. In the second group, users may be attempting to self-diagnose prior to a hospital visit.[1] In this case, queries may be **underspecified** as users may not have the medical expertise to know what to search for, and an interactive interface may be needed to help users explore the options. In fact, a user study reported on the Journal of the American Medical Association [3] has concluded that "using search engines and simple search terms is not efficient."

Our goal is to develop a better search engine and search interface to support users in understanding health information and making health decisions. This work examines the search interface deployed by Healia, a health-related vertical search engine that focuses on the above challenges (i.e. highly personalized/specific results, underspecified queries).[2] The paper is divided as follows: First, we describe Healia's search interface, highlighting the features we believe are important in supporting user interaction and information finding in health. Then, we present results from query log analysis, which show how these advanced features are utilized. Finally, we present our conclusions and thoughts on future work.

2. HEALIA SEARCH INTERFACE

The Healia Search Interface, which can be accessed at http://www.healia.com (a screenshot is shown in Appendix A), consists of five main areas of user interaction: a search box, a personalization filter, faceted browsing, suggested query terms, and entry to Pubmed/Clinical Trials information. We imagine the searcher may use this interface in the following scenario:

[1] [2] also reports that for an increasing number of young users, the Internet is the preferred source to learn about health.

[2] Another major challenge for health search engines is to provide information that is credible and trustworthy. In this paper we focus on the interface aspects and will not discuss how we optimize the Healia search engine to achieve this.

1. Enter query term, e.g. diabetes, and see initial results.
2. Personalize the results with the filter, e.g. click on "Female" and "African American" to return results specific to a demographic. The personalization filter also allows filtering of results based on reading level and accreditation.
3. Explore the various facets of diabetes, which includes "Prevention," "Causes," "Symptoms," "Diagnosis," and "Treatment".
4. Try the suggested query terms, which proposes similar searches and more specific/general medical terms.
5. Further, if the user is determined to understand more, the entry points to Pubmed journal articles and clinical trials information provide a way to sift through expert information.

We can view user interaction with Healia as the following diagram (Figure 1), where the searcher is given one of five actions.[3] Upon choosing an action, the searcher will see a new results page and can continue interacting with the system with different actions until satisfaction.

In the following, we will study user behavior on the Healia website under the framework of these five user actions.

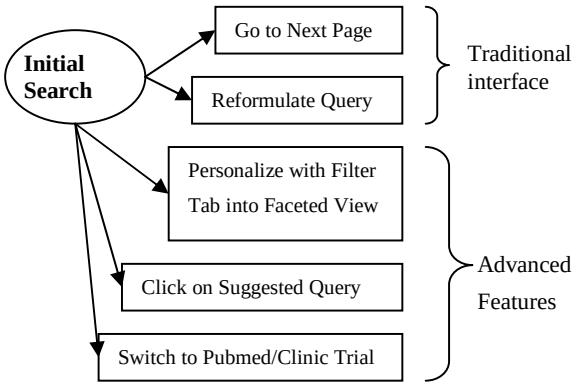


Figure 1: Five possible user actions are available to a searcher on each Healia search results page.

3. QUERY LOG STUDY

We are interested in understanding how searchers use the Healia search interface, in particular, to what extent are the advanced search features used by different types of users. To this end, we mine the query logs to obtain statistics of the five different user actions defined in Figure 1. We filter the log such that only user sessions with one or more actions after the initial search results page are used (i.e. we do not consider cases where the user session ends after a single query and there is no further interaction with the system).

Following the work of [4], we divide our users into regular users and "expert" users, where "expert" is defined by whether the user enters Healia's PubMed search interface to access scientific journal article. Manual inspection of these "expert" search queries reveal many technical terms and PubMed author names, leading us to believe that these searches are meant to pinpoint specific documents and is therefore qualitatively different from the complex and exploratory search tasks of a consumer health user. Among the 6800 unique users in our data, roughly 8% were classified as "expert" under this heuristic.

3.1 What are the most frequent actions taken by users?

First, we measured the frequency of each user action and show the results in Table 1. We observe that the traditional search interface actions of "Reformulate Query" and "Go to Next Page" consists of the majority (82.7%) of user actions and the advanced search features are utilized with less frequency (17.3%) in total. Among the advanced features, "Suggested Query" and "Personalization / Faceted Tab" are used equally often. Interestingly, many user-entered query reformulations are often achievable by personalization filters and tabs, for example:

- "strep throat"(original query) → "strep throat in children" (reformulated query, typed in by user)
- "quit smoking"→ "quit smoking methods"
- "uterine infection"→ "cause of uterine infection"

These query reformulations reflect the need to get more personalized and specific information, which is exactly what can be accomplished by the advanced features, but users often chose to type additional query terms (which is more time consuming). The reason may be that users now are used to the single box search interface.

Table 1 also shows that expert users use advance features roughly 3%-5% more than regular users.

Table 1. Percentage of User Actions

User Action	ALL USERS	EXPERT USERS	REGULAR USERS
Reformulate Query	47.9	43.4	49.5
Go to Next Page	34.8	33.7	35.7
Personalization Filter / Tab into Facet	8.3	12.7	7.5
Click on Suggested Query	7.6	10.2	7.3
Switch to PubMed or Clinical Trials	1.4	-	-

3.2 How long do users interact with the search engine?

Second, we calculated the length of a user session, in terms of the number of user interactions. Long user sessions indicate an extended interaction with the search interface. Figure 2 shows the cumulative density function for user actions: 71% of all user

[3] In this study, we will not examine other Healia features, such as the entry points to the Healia Health Guides (editorial content) and the Healia Communities social support network. Also, we consider personalization filters and faceted tabs as the same type of user action since they both involve filtering the current list of search results.

sessions end after one user action, 81% of all user sessions end with two or less user actions, and 91% of all user sessions end with four or less user actions. The majority of user sessions are short, but there are a significant number of extended interactions.

We also observe that the sessions of expert users are shorter than that of regular users. Two possible explanations are: (1) the search tasks of regular users are more complex and require extended interaction; (2) expert users used advanced search features more often than regular users, thus finding information faster. Further work is needed to test these hypotheses.

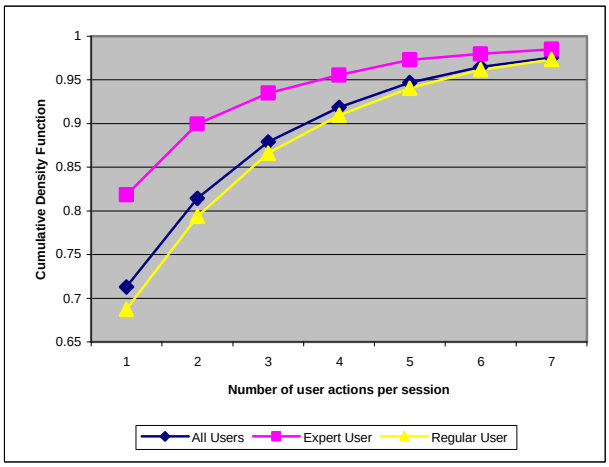


Figure 2: Number of user actions per session. This is a cumulative plot indicating the percentage of user sessions with less than or equal to X user actions.

3.3 How diverse are the actions in each user session?

We are also interested in seeing whether users employ a variety of actions in a user session, since a diversity of actions implies the user's sophistication with the search strategy. We found that users tend to stick to a few actions (possibly due to familiarity with its intended results): Of all the sessions that have at least three actions, 44% involve only one type of action, e.g.:

- reformulate query→ reformulate query → reformulate query
- next page → next page → next page

42% of user sessions involve two types of actions, e.g.:

- reformulate query → next page → reformulate query
- suggested term → personalize → personalize

Only 12% of user sessions involve three or more types of actions.

3.4 What kinds of personalization filters and facets are being used?

Figure 3 indicates facet usage by measuring the percentage of time each facet tab is clicked on in the query log. We find that users are most interested in the "symptoms" facet of their search results, implying that users are indeed using the Internet as a tool for self-diagnosis. In fact, as many as 20% of distinct queries entered in conjunction with faceted tabs contain the words "photo" or "picture" (e.g. "pictures of pink eye", "scabies photo").

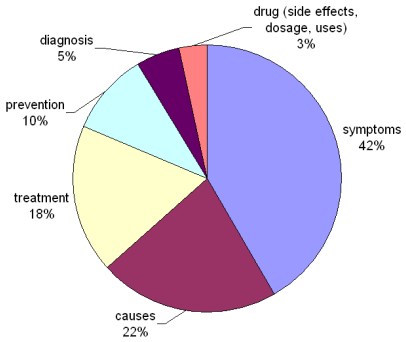


Figure 3: Facet usage. Users are most interested in finding out about "symptoms" (42%), "causes" (22%), and "treatments" (18%) of diseases.

Figure 4 shows the percentage of time each type of personalization filter is used. Users most often filter results by "gender" and "age." The more popular setting for the gender filter is "female" (68%); for the age filter, the breakdown is "kids" (57%), "teens" (26%), "seniors" (17%). These statistics may have interesting implications as to who may be the main consumers of Internet health information (i.e. women and parents).

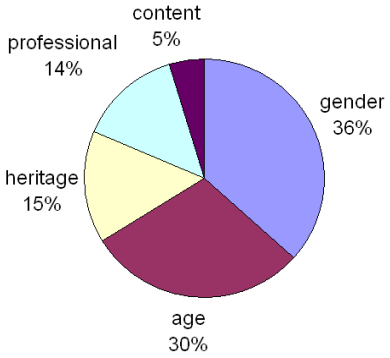


Figure 4: Personalization filter usage. Users filter results most often by gender (female/male) and age (kids/teens/senior), followed by heritage (African/Asian/Hispanic/Native), professional, and content (e.g. easy to scan, interactive tools)

4. SUMMARY AND PROSPECTS

We have advocated that a vertical search engine for health should provide features that support the complex information need of users, which can be *highly personalized*, *highly specific*, and *underspecified*. Consumer health search is an "exploratory search" problem [5] where users are "searching to learn." Our query log study of the Healia search interface found that:

1) Users sometimes opt to use the traditional single search box paradigm even when advanced features provide one-click solutions to personalization and more specific information. Nevertheless, we observe a promising ~17% usage of advanced features on Healia.

2) Expert user sessions are shorter than those of regular users. It is not yet clear whether this is due to simpler information need for technical searches, or faster task completion since experts use more advanced features.

3) User interactions with the search interface are not very diverse, with only 12% of user sessions involving three or more actions.

4) The most commonly-used facet is "symptoms", implying an audience that uses health search for self-diagnosis. Commonly-used filters are gender and age.

We are interested in the following open questions:

- How do we design search interfaces so that advanced search features can be easily learned and adopted?
- What other advanced search features are useful in helping consumer health users make informed health decisions?

Regarding the first point, it has been shown by [6] that a user who learns a good search strategy performs significantly better in retrieving domain-related information. Further, [7] presents design recommendations for making faceted search, in particular, more effective.

We have recently built a new version of the Healia search interface, which includes federated search (of the Web, PubMed, and Clinical Trials), a more streamlined presentation of filters and tabs, and significant improvements in response time for user interactions. We believe these enhancements will further improve the user experience; it would be interesting to perform a comparative study of query logs between these two versions for evaluation purposes.

5. REFERENCES

[1] Forrester Research. (2006). North American Consumer Technology Adoption Study (NACTAS) Q4.

[2] Hanson, J. (2007). How Different Generations Use Online Health Research. Forrester Research.

[3] Berland, G.K., et. al. (2001). Health Information on the Internet—Accessibility, Quality, and Readability in English and Spanish. Journal of the American Medical Association.

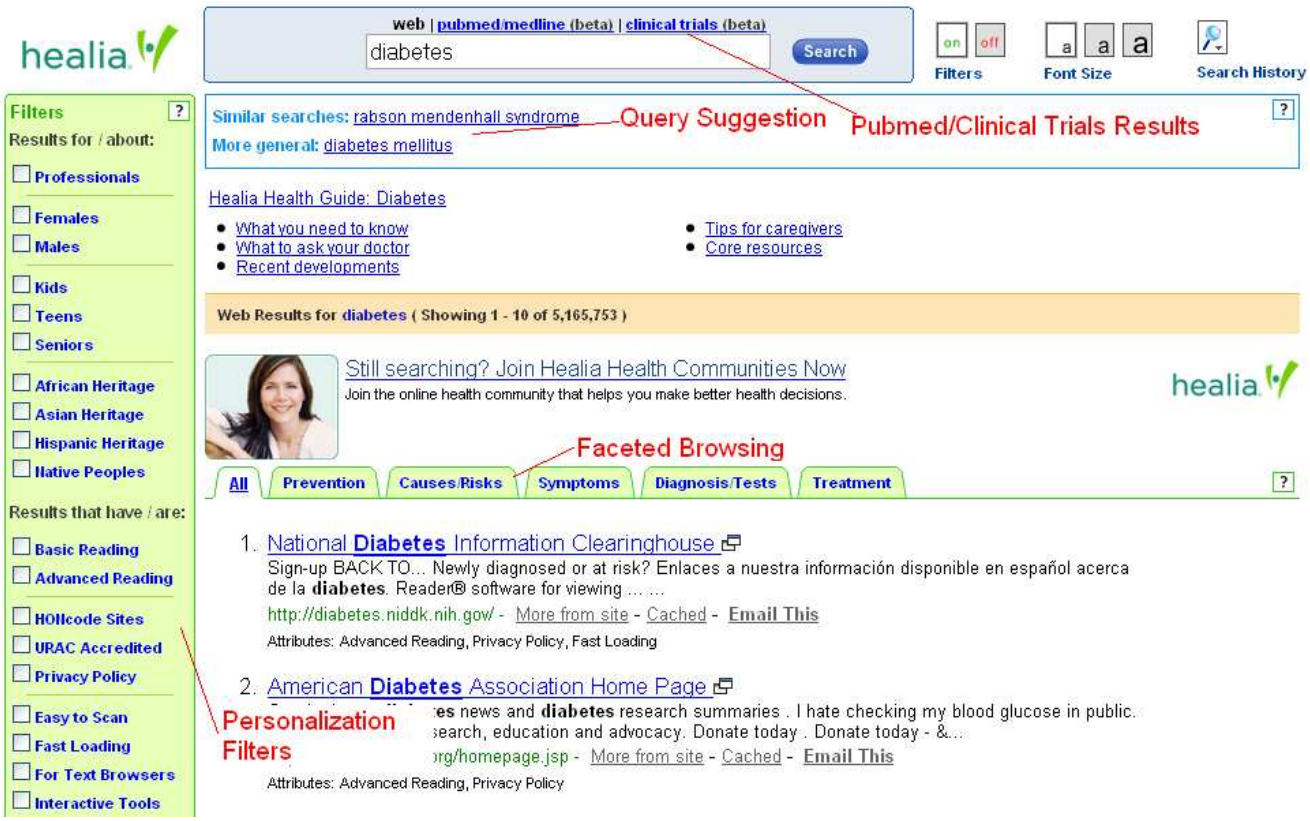
[4] White, R., Dumais, S., Teevan, J. (2008). How Medical Expertise Influences Web Search Interaction. SIGIR.

[5] Marchionini, G. (2006). Exploratory search: From finding to understanding. Communications of the ACM, 49 (4).

[6] Bhavnani, S.K. (2002). Domain-specific search strategies for the effective retrieval of health and shopping information. SIGCHI.

[7] Hearst, M. (2006). Design recommendations for hierarchical faceted search interfaces. SIGIR Faceted Search Workshop.

Appendix A: Healia Search Engine Interface, August 2008 (http://www.healia.com)



Collaborative Query Term Suggestion

Gene Golovchinsky
FX Palo Alto Laboratory, Inc.
3400 Hillview Ave, Bldg 4.
gene@fxpal.com

Pernilla Qvarfordt
FX Palo Alto Laboratory, Inc.
3400 Hillview Ave, Bldg 4.
pernilla@fxpal.com

Jeremy Pickens
FX Palo Alto Laboratory, Inc.
3400 Hillview Ave, Bldg 4.
jeremy@fxpal.com

ABSTRACT
Query term suggestion has been an important component of information seeking support tools. It has been used for automatic query expansion and re-ranking operations as part of relevance feedback, manually during exploratory search, and interactively through user selections of suggested terms. Term suggestion has been driven by document analysis and through collaborative filtering algorithms. In this work, we describe a novel approach to generating query term suggestions based on activities of a coordinated search team. Terms extracted from documents based on the actions of one team member and suggested as possible query terms to another member. We evaluated the effectiveness of this approach and found a significant correlation between the use of suggested terms and improvements in recall.

Categories and Subject Descriptors
H.3.3 [**Information Search and Retrieval**]: Relevance Feedback, Query formulation; H.5.3. [**Group and Organization Interfaces**]: Computer-supported cooperative work.

General Terms
Experimentation, Human Factors.

Keywords
Collaborative Information seeking, information retrieval, query reformulation, query expansion

1. INTRODUCTION
One of the challenges of information seeking is to translate latent, perhaps poorly understood information needs into specific queries that will be effective at retrieving relevant documents. Typically users generate search terms that are used to retrieve documents, the reading of which may inspire the user to think of additional query terms that retrieve more documents, et cetera. While this process can be successful in some cases, it is only as good as a user's ability to generate query terms. This can vary based on a user's experience in searching in general, and also based on familiarity with a specific topic.

Automatic query expansion approaches are well known in the literature [9]. Typically using an approach like relevance feedback[10], the system can automatically extract terms from documents a user has marked as relevant and add those terms to the previous set to form a new query. Relevance feedback information can be collected explicitly in the form of judgments, implicitly through various actions such as link selection [1] or annotation [5], or through collaborative filtering based on similar patterns of behavior (e.g., Amazon.com).

One problem with relevance feedback is its opacity. Koenemann [6] found that subjects performed better with (and had higher preference for) interfaces that showed suggested query terms rather than performing automatic relevance feedback. Results from another study [1] found that subjects were really interested in controlling which suggested terms are used in subsequent queries, but that they were not interested in the mechanisms of generating the term suggestions.

In this paper, we describe and evaluate a technique for suggesting potentially-useful query terms in the framework of collaborative information seeking [7]. We have previously shown that teams of people working together on a shared information need perform more effectively and more efficiently than individuals whose results are pooled after the fact [8].

One of the ways in which our system supports collaboration is by offering to one team member suggestions of potentially useful query terms based on relevance judgments made by the other team member. In the rest of this paper, we first give an overview of our collaborative search system, and then describe an evaluation of the term suggestion algorithm.

2. COLLABORATIVE SEARCH
We built a collaborative search system called Cerchiamo [8] to explore various aspects of collaborative information seeking. The system allows two people to work together to find information related to a shared information need. The two collaborators assume the roles of Prospector and Miner: the Prospector identifies promising queries and evaluates the initial portion of the results list; the Miner makes additional judgments of relevance on documents retrieved (but not seen by) the Prospector. In addition, the system identifies terms characteristic of relevant documents (as judged by the Miner) and makes them available for incorporating into subsequent queries at the Prospector's discretion.

Thus there are two asynchronous data flows during a search session: documents move from the Prospector to the Miner, and potentially useful query terms move from the Miner to the Prospector. In each case, a ranked list of objects is maintained by the system based on inputs from both users.

To understand how the suggested query term list is created, we must first understand how the ranked list of documents that the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
HCIR'08, October 23, 2008, Seattle WA
Copyright 2008 ACM 1-58113-000-0/00/0008…$5.00.

Miner operates on is created. As the Prospector works, he issues queries, thereby generating multiple ranked lists of documents L returned in response to those queries. For each query k and ranked document list returned by that query L_k, the mediation algorithm computes two weighting variables: relevance $w_r(L_k)$ and freshness $w_f(L_k)$.

$$w_f(L_k) = \frac{|\, unseen \in L_k \,|}{|\, seen \in L_k \,|}$$

$$w_r(L_k) = \frac{|\, relevant \in L_k \,|}{|\, nonrelevant \in L_k \,|}$$

The query freshness weight w_f is the ratio of unseen (retrieved by the engine, but not yet manually examined) to seen (retrieved and manually examined) documents in L_k. The query relevance weight w_r is the fraction of seen documents that were judged relevant for that query. These two factors are designed to counter-balance each other: queries that retrieve many relevant documents get high relevance weights, but once most of the retrieved documents have been seen, the list's overall freshness goes down. Similarly, another query which has retrieved relatively fewer relevant documents would still receive a higher freshness weight if most of the retrieved documents had not yet been examined. As the system runs, these weights are updated continuously based on activities of the searchers.

While these weights can be used to rank documents for relevance judgments by the Miner [8], they can also be used to rank query terms associated with relevant documents as follows: Let $rlf(t, L_k)$ be the number of documents in query result list L_k in which term t is found. We call this the "Ranked List Frequency". We now define the score for term t as the sum over all ranked lists of the weighted rlf score:

$$score(t) = \sum_{L_k \in \{L\}} w_r(L_k) w_f(L_k) rlf\,(d, L_k)$$

This formula selects terms associated with promising (high w_r) and relatively unexplored (high w_f) queries, and prefers terms that occur in multiple sets of search results via summation over all lists.

This allows Prospector and the Miner to work independently: while the Prospector issues a new query, the Miner is making relevance judgments on documents retrieved by earlier queries.

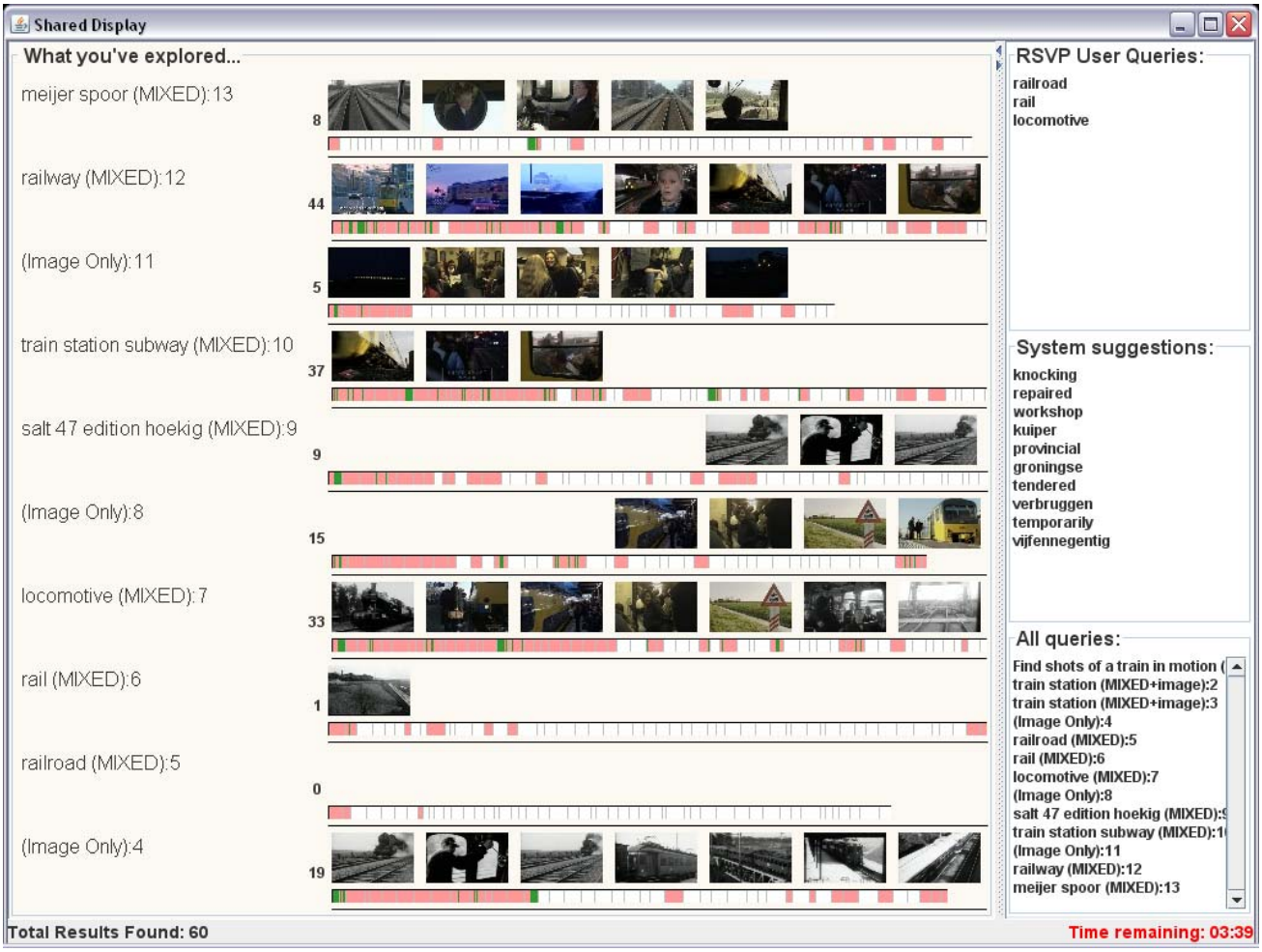


Figure 1. Shared display showing suggested query terms in the middle on the right. Bars below images represent shots judged relevant (green), non-relevant (pink), or not judged (white)

As more judgments of relevance accrue to some queries, terms found in documents returned by those queries are boosted in the suggestion list. This is one of the important concepts in this work: By allowing the users to work separately but with synchronous mutual influence, the Miner is able (through system mediation) to discover query suggestions that the Prospector had overlooked.

In Cerchiamo, the Prospector and Miner use user interfaces specifically designed to support their roles. However, the team also shares an additional interface (Figure 2) that shows a history of queries, associated shots, and histograms of relevant/non-relevant/not judged shots. In addition, it shows queries generated by the Miner directly (RSVP User queries), and the system-suggested query terms. The purpose of the shared display was not only to show the system suggested terms, but to give the team a shared understanding of their progress during the search session.

Figure 2. Cerchiamo team at work

3. EVALUATION

As part of the TRECVid 2007 competition, we performed experiments using our collaborative search system [1]. Two-person teams were asked to identify as many relevant documents for each given topic as possible in 15 minutes. Searches were performed on 24 topics in total. The material being searched consisted of Dutch television programs, and included textual transcripts that were generated by Dutch speech-to-text conversion followed by automatic translation into English. Not all terms were translated: some Dutch words remained in the corpus. For example, the term list in Figure 2 includes English words such as "knocking," "workshop," and "temporarily," and Dutch words "kuiper," "verbruggen," and "vijfennegentig."

The two team members assumed the two roles of Prospector and Miner. The Prospector used an interface similar to our system from previous TRECVid competitions [3], and a Miner used an RSVP-style interface designed to facilitate relevance judgments on a queue of images. The team members were seated next to each other as illustrated in Figure 2. Participation in TRECVid gave us access (after the competition) to the ground truth for each query, based on which various aspects of the system could be evaluated.

Among other measures, we assessed the utility of these suggested terms. We found that the Prospector used on average unique 1.7 system suggested terms per topic (SD=1.94). In comparison to the average number of queries per topic (21.8 queries, SD=5.59), this may not seem like a very high number, but when investigating its effect on the overall performance, we found that the use of suggested terms significantly correlated with recall ($r(20)=0.43$, $p<0.05$). The more system suggested terms the Prospector used, the higher recall the team achieved. This means that the Miner's actions, mediated by the system, influenced the Prospector's behavior, and as a result, the team performed better.

Next we investigated how the system suggested terms were used for different kinds of topics. We divided the search topics into two groups, sparse and plentiful, based on the number of relevant documents in the corpus. The plentiful group contained topics with at least 130 documents, and had on average 332 relevant documents in the corpus. The sparse group had on average 60

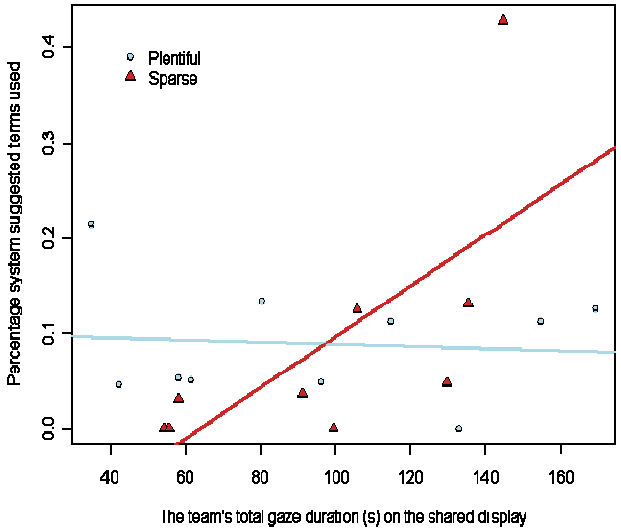


Figure 3. Relation between percentage system suggested terms used and the team's total gaze on the shared display.

relevant documents each in the corpus. The number of terms used in sparse and plentiful topics did not differ ($t(20)=.156$, *ns.*). We found that for plentiful topics there was no correlation between recall and the number of system suggested terms used ($r(9)=.32$, *ns.*). For the plentiful topics, we found a near significant correlation ($r(9)=.55$, $p=.08$) between recall and number of system suggested terms used. This result indicate that for sparse topics, topics where the team had trouble finding documents, the system suggested terms helped them finding new queries that could potentially open up new avenues for exploration.

In addition, we looked at how much time the team as a whole spent looking at the shared display. We used that gaze time of the team as whole since the Miner might look at the system suggested terms and encourage the Prospector to use some of the terms. Interestingly, we found that for the two different kinds of topics the team members used the display differently. As Figure 3 shows, for sparse topics, the number of system suggested terms used goes up the more the team members look at it ($r(7)=0.69$, $p<0.05$), indicating that for these topics the terms were looked at and used. However, the relation between total gaze duration on the shared display and system suggested terms was not found for plentiful topics ($r(8)=-0.08$, *ns.*) indicating that for these topics the team either utilizes other information from the shared display, such as the history of the queries and their performance, or does

not need to utilize system suggestions. For the plentiful topics, each query returned more relevant results so there was less need for assistance with query formulation, while the need for reminders of the search field already covered was higher.

Together these results indicate that system suggested terms as implemented in Cerchiamo was useful for the team performance, in particular for topics with few relevant documents to be found. Interestingly, as the teams spend more time looking at the information on the shared display, including the system suggested terms, while working on sparse topics the more terms they used, and the more benefit from the terms they gained. However, this was not true for the plentiful topics, possibly indicating that the teams were not as careful in selecting which system suggested terms to use to gain as much as possible from their use.

4. CONCLUSIONS

We described an algorithm for identifying promising query terms in search results collected over multiple queries based on judgments of relevance. This technique was used to generate term suggestions for a member of a team engaged in collaborative search activity. The use of these suggested terms correlated with increased recall. This is just a first step in an exploration of system mediation for collaborative information seeking. This is an emerging inter-disciplinary field that will benefit from contributions from CSCW, HCI and IR communities.

5. REFERENCES

[1] Adcock, A., Pickens, J., Cooper, M., Chen, F., and Qvarfordt, P. FXPAL Interactive Search Experiments for TRECVID 2007. In *Proceedings of TRECVid 2007*. March 2007. Available on the web at http://www-nlpir.nist.gov/projects/tvpubs/tv7.papers/fxpal.pdf

[2] Belkin, N.J., Cool, C., Head, J., Jeng, J., Kelly, D., Lin, S.J., Lobash, L. Park, S.Y., Savage-Knepshield, P., and Sikora, C. Relevance feedback versus Local Context Analysis as term suggestion devices: In *Proceedings of the Eighth Text Retrieval Conference TREC8*. pp. 565. 2000. Available online at http://trec.nist.gov/pubs/trec8/papers/ruint.pdf

[3] Girgensohn, A., Adcock, J., Cooper, M.D., and Wilcox, L. A synergistic approach to efficient interactive video retrieval. In *INTERACT*, pages 781-794, 2005.

[4] Golovchinsky, G. Queries? Links? Is there a difference? In *Proceedings of CHI'97*, ACM Press, pp. 407-414. 1997

[5] Golovchinsky, G., Price, M.N., and Schilit, B. From Reading to Retrieval: Freeform Ink Annotations as Queries. In *Proceedings of SIGIR99, ACM Press, pp. 19-25.* 1999

[6] Koenemann, J. *Relevance feedback: usage, usability, utility.* Ph.D. Dissertation. Rutgers University, Dept. of Psychology. New Brunswick, NJ. 1996.

[7] Pickens, J. and Golovchinsky, G. Collaborative Exploratory Search. In *Proceedings of HCIR07*, October 23, 2007. pp. 21-22. available online at projects.csail.mit.edu/hcir/web/hcir07.pdf

[8] Pickens, J., Golovchinsky, G., Shah, C., Qvarfordt, P., and Back, M. Algorithmic Mediation for Collaborative Exploratory Search. In *Proceedings of SIGIR08*, ACM Press, pp. 315-322. 2008.

[9] Robertson, S. and Spärck Jones, K. Simple, Proven Approaches to Text Retrieval. University of Cambridge, Technical Report UCAM-CL-TR-356, December 1994.

[10] Salton, G. and Buckley, C. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41 (4), pp. 288-297. 1990.

Lightweight Additions to the Web Search Interface Supporting Exploratory Web Search

Orland Hoeber
Department of Computer Science
Memorial University of Newfoundland
St. John's, NL A1C 3X5
Canada
hoeber@cs.mun.ca

ABSTRACT

In this paper, the features of *TheHotMap.com* that support exploratory Web search processes are described. This system grew out of two academic research projects that explored the use of visualization and interaction as a means for supporting users as they conduct Web search tasks. In *TheHotMap.com*, three lightweight interface extensions have been added to the commonly used list-based representation of Web search results. These can be used independently or together to support users as they craft queries and explore search results. A scenario of using the system for exploratory Web search is described in this paper; a live demonstration will be provided at the workshop.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces

General Terms

interfaces, search, exploration

Keywords

Web search, information visualization, interaction

1. INTRODUCTION

In recent years, Web search has become an important part of the information-seeking and knowledge-generating activities of the general public. A study from 2004 reported that 88% of Web users start their information-seeking tasks with a search engine [13]. Although more recent studies on Web user behaviour do not directly address the frequency of use of search engines, market research has shown monthly increases in their use in the United States [3].

Although searching has become the primary tool for finding information on the Web, the interfaces employed by the top search engines have changed very little since the early days of Web search. The primary interface features continue to be a query box for capturing the searcher's intent, and a list-based representation of the search results. Although such interfaces are very easy to learn and use, their power for supporting complex or exploratory search tasks is limited.

Our primary motivation for this research has been to explore the use of information visualization and interaction techniques to support Web search activities. Information visualization is a technique for creating interactive graphical representations of abstract data or concepts [15]. Moreover, information visualization promotes a cognitive activity in which users are able to gain understanding or insight into the data being graphically displayed by taking advantage of human visual information processing capabilities [14].

The potential benefits of employing information visualization and interaction techniques to support Web search activities are immense. However, the challenge is to show restraint in the design of such systems, and avoid overly complex visual representations and interaction methods that are difficult to learn and use. Our focus in this paper is on three lightweight extensions to the commonly used list-based representation that support exploratory Web search activities.

2. RELATED WORK

This work is closely related to our previous research activities in the development of visual and interactive interfaces for Web search. In particular, the system is based on a combination of two of our previous research prototypes: HotMap [8] and Wordbars [7]. These prototypes were originally developed with the purpose of exploring visual representations, interaction, and use of various types of information to support Web search activities. Combined together, they allow the searcher to easily switch between their two primary tasks of interactive query refinement and interactive search results exploration [9].

As research tools, these prototypes were useful for validating the potential utility of the proposed techniques [10, 11]. However, they were not designed for public release. *TheHotMap.com* is a complete re-implementation and extension of the methods employed by these previous works.

Others have explored the use of visual interfaces to support the evaluation of Web search results. Heimonen and Jhaveri [6] created an icon-based representation of the locations of specific query terms within individual search results sets. Based on TileBars [5], this system allowed the searcher to see where in the resulting documents their search terms

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
HCIR 2008 Redmond, WA, USA
Copyright 200X ACM X-XXXXX-XX-X/XX/XX ...$5.00.

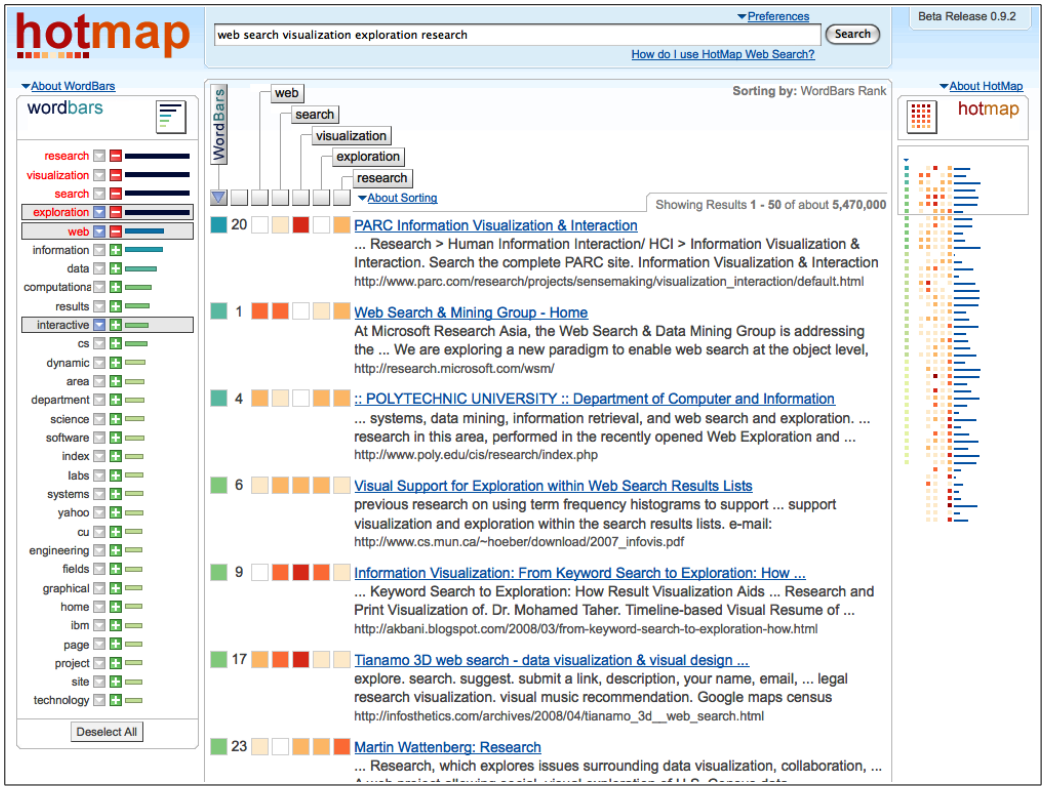


Figure 1: A screenshot of *TheHotMap.com*. Note the lightweight controls representing common terms in the search results set (WordBars histogram, on the left) and the frequency of the query terms in the search results set (HotMap overview, on the right), along with the re-sortable search results list using the HotMap query term headers (centre).

were being used together.

In VIEWER [1], the frequency of all combinations of the query terms were counted within the document surrogates. This information was presented in a histogram representation. Selections within the histogram allowed the searcher to filter the search results set based on specific combinations of the query terms.

Web search clustering systems, such as Clusty [2] and Grokker [4], dynamically identify and label clusters of documents discovered within the search results sets. Normally presented in a tree-based structure, users can expand and select clusters, resulting in a filtering of the search results set. Kules [12] extended the standard paradigm for clustering search engines by providing a consistent naming scheme for the clusters. The result is a system that allows users to learn the names and meanings of the clusters over time.

3. SYSTEM FEATURES

TheHotMap.com is implemented as a Web search interface layer overtop of the search results provided by the Yahoo API [16]. There are three main features that support exploratory Web search activities using the system: the WordBars histogram, the HotMap overview of the full search results set, and the re-sortable search results list using the HotMap query term headers. Each of these features are described in more detail below; specific details on the techniques and their potential benefits are provided in [10, 11].

Figure 1 provides a screenshot of *TheHotMap.com*. Although the number of daily queries is currently limited, the system is available as a publicly accessible demonstration at `http://www.thehotmap.com/`.

3.1 WordBars Histogram

The WordBars histogram provides a visual representation of the most frequently appearing terms within the search results set, allowing the relative frequency of these terms to be easily observed. Users can interactively re-sort the search results set by selecting the arrow icon beside any term of interest. A visual indicator within the search results list (under the vertical WordBars button) depicts the frequency of the selected terms within each search result. Searchers can easily select and un-select terms of interest as they explore the search results. Interactive query refinement is supported by clicking the plus icon beside any term users wish to add to their queries, or the minus icon beside any term users wish to remove from the query.

3.2 HotMap Overview

The HotMap overview provides a compact visual representation of the entire set of search results that are present in the list-based representation. In the current implementation, the system collects 50 search results per page. Colour coding is used to represent the frequency of the query terms within the search results set; bars that are relative to the length of each search result title are included to support the visual mapping between the search results set and the

HotMap overview. The colour coding of term frequencies is also used in the search results list, resulting in the HotMap overview appearing as a "zoomed out" view of the search results set.

The HotMap overview supports the visual exploration of the search results. As users identify documents of interest, they may click on the abstract representation of the search result in the HotMap overview to cause the search results list to scroll to that location. The system temporarily highlights the corresponding search result that was selected in the HotMap overview, allowing users to easily relate their selection in the overview to the scrolled location in the search results list.

3.3 HotMap Re-Sorting

In addition to the re-sorting supported via the WordBars histogram, searchers may also re-sort the search results based on the frequency of use of their specific query terms within the search results. Clicking on any of the query term headers above the search results list will cause the search results to be re-sorted. Although the default sorting method is to perform single-term sorting, an advanced feature is available that supports nested sorting.

4. EXPLORATORY SEARCH SCENARIO

A scenario illustrating the use of *theHotMap.com* when conducting an exploratory Web search based on incomplete knowledge about the task is provided in Figure 2. This scenario shows how a user can start with an initial query (a) and use the features of the WordBars histogram to explore the search results and learn about the topic (b and c). The WordBars histogram also supports the user in making modifications to the query based on what they have learned (d). The HotMap overview allows the searcher to visually inspect areas of interest in the search results set and easily jump to the corresponding location in the search results list (e). The system also supports re-sorting the search results based on the importance the searcher places on their query terms (f).

Although this scenario shows the searcher first using the WordBars histogram features, followed by the HotMap overview and re-sorting features, this order of use is not enforced by the system. Searchers are free to use whichever feature of the system that best supports their current search objective. For example, if the searcher wishes to start with a somewhat vague initial query, and then explore and evaluate the search results seeking relevant terms to add to their query, they may do so easily using the WordBars histogram features. Alternately, if the searcher is already confident in the quality of their query and they wish to explore the search results seeking relevant documents, they may do so visually using the HotMap overview, or via interactive re-sorting of the search results based on their query terms.

5. CONCLUSIONS

TheHotMap.com adds three lightweight additions to the commonly used list-based representation of Web search results. Used together or separately, the features supported by these additions provide flexible methods for conducting exploratory Web search activities, allowing users to interactively refine their queries and interactively explore the search results. Visualization techniques are used to depict information that is relevant to the searchers' primary tasks and goals. The interactivity of the system allows searchers to take an active role in their Web search activities.

6. ACKNOWLEDGMENTS

The authors thank the University-Industry Liaison Office at the University of Regina for their support of this work.

7. REFERENCES

[1] E. Berenci, C. Carpineto, V. Giannini, and S. Mizzaro. Effectiveness of keyword-based display and selection of retrieval results for interactive searches. *International Journal on Digital Libraries*, 3(3):249–260, 2000.

[2] Clusty. Clusty: the clustering search engine. http://www.clusty.com/, 2008.

[3] comScore. comScore releases June 2008 U.S. search engine rankings. http://www.comscore.com/press/release.asp?press=2337, June 2008.

[4] Grokker. Grokker - enterprise search management and content integration. http://www.grokker.com/, 2008.

[5] M. Hearst. Tilebars: Visualization of term distribution information in full text information access. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, 1995.

[6] T. Heimonen and N. Jhaveri. Visualizing query occurrence in search result lists. In *Proceedings of the International Conference on Information Visualization*, 2005.

[7] O. Hoeber and X. D. Yang. Interactive Web information retrieval using WordBars. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, 2006.

[8] O. Hoeber and X. D. Yang. The visual exploration of Web search results using HotMap. In *Proceedings of the International Conference on Information Visualization*, 2006.

[9] O. Hoeber and X. D. Yang. A unified interface for visual and interactive Web search. In *Proceedings of the IASTED International Conference on Communications, Internet, and Information Technology*, 2007.

[10] O. Hoeber and X. D. Yang. Evaluating WordBars in exploratory Web search scenarios. *Information Processing and Management*, 44(2):485–510, 2008.

[11] O. Hoeber and X. D. Yang. HotMap: Supporting visual exploration of Web search results. *Journal of the American Society for Information Science and Technology*, in press.

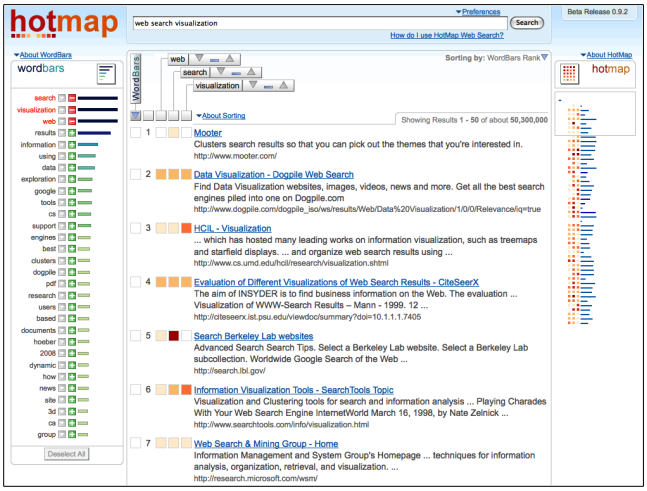
[12] B. Kules, J. Kustanowitz, and B. Shneiderman. Categorizing Web search results into meaningful and stable categories using fast-feature techniques. In *Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries*, 2006.

[13] J. Nielsen. When search engines become answer engines. http://www.useit.com/alertbox/20040816.html, August 2004.

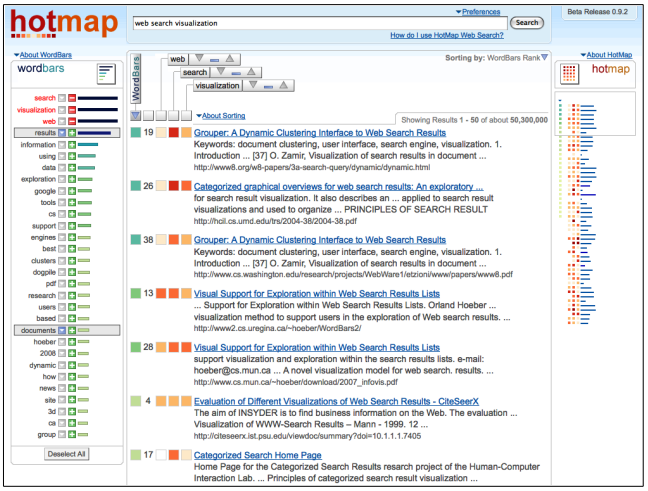
[14] R. Spence. *Information Visualization: Design for Interaction*. Pearson Education, 2nd edition, 2007.

[15] C. Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann, 2004.

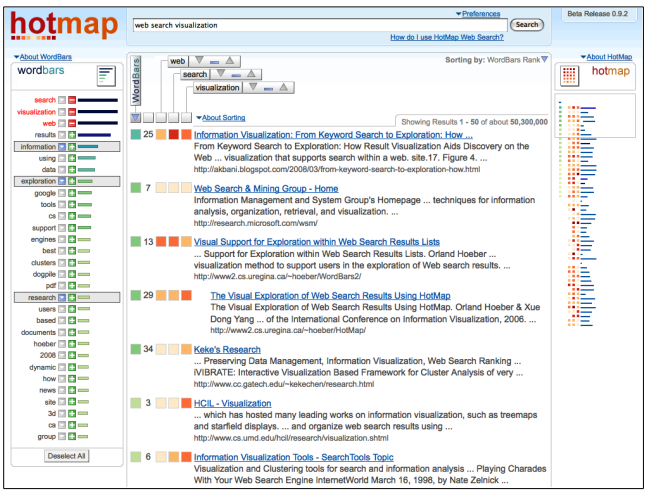
[16] Yahoo. Yahoo search Web searvices. http://developer.yahoo.com/search/, 2008.



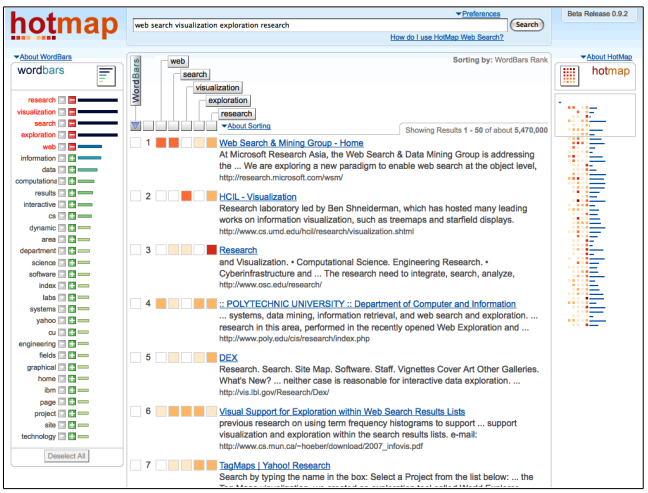
(a) Initial search for *"web search visualization"*.



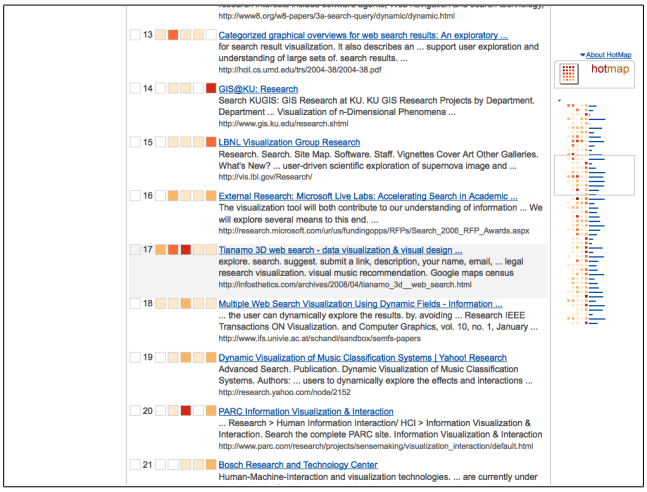
(b) Exploration of the search results by selecting *"results"* and *"documents"* from the WordBars histogram. Note the highlighted terms and re-sorted search results.



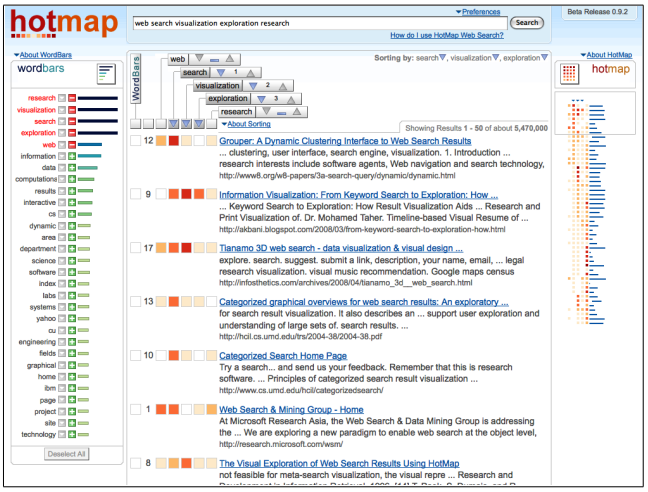
(c) Further exploration of the search results by selecting *"information"*, *"exploration"*, and *"research"* from the WordBars histogram. Note the different order of the search results from the previous step.



(d) The query is refined by adding *"exploration"* and *"research"* using the WordBars histogram. Note the new set of search results, the new WordBars histogram, and the new HotMap overview.



(e) Visual inspection of the HotMap overview reveals a potentially interesting document deep in the search results list. Clicking on it scrolls the search results to the appropriate location, and temporarily highlights the document (doc. 17).



(f) Re-sorting the search results based on query terms of specific interest to the searcher allows them to provide supplemental information about the importance of their query terms to the search process.

Figure 2: Screenshots from *TheHotMap.com* illustrating the features that support exploratory Web search.

Viewing Searching Systems as Learning Systems

Bernard J. Jansen
College of Information Sciences and Technology
The Pennsylvania State University
University Park, PA, 16801, USA
jjansen@acm.org

ABSTRACT
Investigating whether users of a searching system are engaged in a learning environment, the results of this research show that information searching is a cognitive learning process with unique searching characteristics specific to particular learning levels. In a laboratory experiment, we studied the searching characteristics of 72 participants engaged in 426 searching tasks. We developed the searching tasks according to Anderson and Krathwohl's categories of the cognitive learning domain. Research results indicate that *applying* and *analyzing*, the middle two of the six categories, generally take the most searching effort in terms of queries per session, topics searched per session, and total time searching. The lowest two learning categories, *remembering* and *understanding*, exhibit searching characteristics similar to the highest order learning categories of *evaluating* and *creating*. These results suggest that users applied simple searching expressions to support their higher level information needs. These findings points to the need for searching system features that engage the user in a learning process.

Categories and Subject Descriptors
H.3.3 [1] Information Search and Retrieval – *Search process.*

General Terms
Experimentation, Human Factors

Keywords
Information searching, Bloom's Taxonomy

1. INTRODUCTION
In this research, we use learning theory to investigate information searching, which is the process of a user engaging an information retrieval system. Specifically, we aim to discover an inferential framework based on learning theory for indentifying the cognitive category of a searcher's need based on characteristics of the information searching process. From this knowledge, one can then design searching systems to support this specific category of need.

A widespread paradigm for analyzing Web searching is problem solving or decision-making. Donohew and Tipton [4, p. 251] state that information seeking research is intertwined with decision making. Much information and Web searching research is linked with this view of searching as a decision making process.

The recognition of problem solving as a conceptual framework for information searching is not universally accepted. Sperber and Wilson [12] argue that problem solving does not apply to all information searching situations. More importantly, there is a notable lack of empirical data to support the relationship between information searching and problem solving. Most of the published works that discuss the relationship between decision-making and information searching are descriptive in nature (i.e., the proposed decision-making model is not predictive). Few, if any, laboratory studies have linked information searching behaviors with decision-making currently exist [3].

Having a workable framework for information searching is beneficial for designing systems and interfaces to support the process. We therefore explored other possible frameworks in which to view Web searching, most notably as a learning activity.

In this paper, we present a brief literature review, our research questions, research results, and implications for future Web searching systems.

2. REVIEW OF LITERATURE
There is information searching literature that refers to an on-going learning process while a person is engaged in information searching [c.f., 7]. Tang [13] analyzed the searching behaviors of 41 public library patrons and categorized them into two groups based on their exhibited searching strategies, resource-oriented and query-oriented. The resource-oriented searchers made only minor changes to their initial queries. The query-oriented users exhibited a lot of query reformulation. The researcher suggested that there was a learning process inherent in information searching. Halttunen [5] studied whether there were relationships between learning style, academic domain, and teaching information retrieval techniques. The researcher reported that learning styles generated differences in conceptions of information retrieval understanding. The students who were primarily concrete learners reported computer skills and information retrieval methods as important. Students who were reflective learners viewed information retrieval as the knowledge of information needs analysis, methods, and assessment.

However, there has been little research into how or even if learning explicitly manifests itself in the searching process. Bloom's Taxonomy may be a method for investigating Web search as a learning process. Bloom's Taxonomy is a primary classification of learning in the cognitive domain [2]. An updated version, Anderson and Krathwohl's Taxonomy [1, p. 67-68], redefined Bloom's original classifications [1]. Anderson and Krathwohl's Taxonomy is a six-tiered model for classifying learning according to cognitive levels of complexity.

We conducted a laboratory study to investigate learning as a framework for understanding information searching, with full results reported in [6].

3. RESEARCH QUESTIONS
Our research question is: *Is a learning paradigm effective for analyzing information searching?*

Table 1. Anderson and Krathwohl's Taxonomy with Searching Scenarios

Classification	Definition	Example Scenario
Remembering	Retrieving, recognizing, and recalling relevant knowledge from long-term memory	List 5 movies directed by Steven Spielberg.
Understanding	Constructing meaning from oral, written, and graphic messages through interpreting, exemplifying, classifying, summarizing, inferring, comparing, and explaining	Give a brief plot summary of the TV show, Veronica Mars.
Applying	Carrying out or using a procedure through executing, or implementing	What are some possible characteristics of a person who would enjoy trip-hop music?
Analyzing	Breaking material into constituent parts, determining how the parts relate to one another and to an overall structure or purpose through differentiating, organizing, and attributing	A certain television show contains intense violence and coarse language. Which rating should it receive?
Evaluating	Making judgments based on criteria and standards through checking and critiquing	Create a list of pros and cons for the new iPod Shuffle. Based off of this, would you purchase it (assuming you had the money)? Why or why not?
Creating	Putting elements together to form a coherent or functional whole; reorganizing elements into a new pattern or structure through generating, planning, or producing	Which do you think will have better overall sales -- the XBox 360, the Nintendo Wii, or the Playstation 3? Why?

Hypothesis 1. *There will be a significant difference in the number of queries per session among the classifications in Anderson and Krathwohl's taxonomy.*

Hypothesis 2. *There will be a significant difference in the number of topics per session among the classifications in Anderson and Krathwohl's taxonomy.*

Hypothesis 3. *There will be a significant difference in the duration of sessions among the classifications in Anderson and Krathwohl's taxonomy.*

These hypotheses focus on the query or series of queries. Although an acknowledged imprecise representation of the underlying information need [3], the query is the central aspect of information searching and information retrieval [10, 14]. Numerous empirical studies have focused on the various aspects of the query as surrogates for the expression of need, including session length [9], number of terms [15], and use of keywords [16]. Therefore, we believe the number of queries per session, topics in the session, and session duration are appropriate searching characteristics for this study. We define a session as the series of interactions between the searcher and information system(s) while addressing one of the given searching scenarios.

Using Anderson and Krathwohl's taxonomy of learning in the cognitive domain, we developed searching tasks for each of the taxonomy's six categories. We then analyzed the exhibited searching characteristics to detect differences in searching behavior among them.

4. METHODS

We constructed searching scenarios for each level in Anderson and Krathwohl's Taxonomy, with each scenario correlated to one classification. The searching scenarios were pilot tested twice before we used them in a laboratory study. The six classifications with definitions and example searching scenarios are shown in Table 1. Seventy-two subjects participated in a laboratory study. Each participant engaged in six searching scenarios and were instructed to address the scenarios. Each participant had access to an individual computer with Internet access. All user interactions with the computer were logged using a non-intrusive logging software package. We analyzed participant interactions in accordance with standard characteristics of information searching using transaction log analysis as the methodological approach.

5. RESULTS

We investigated whether or not there would be a significant differences in (1) the number of *queries per session,* (2) number of *topics per session*, and (3) the *duration of session* among the classifications in Anderson and Krathwohl's Taxonomy. A topic is the information focus of one or more queries. A searching session may have several topics.

For number of queries per session, we used a one-way ANOVA statistical analysis to compare means and variance among the classifications. The one-way ANOVA tests whether two or more groups are significantly different. Our results indicate that there is a significant difference among the groups ($F(5) = 5.778$, $p < 0.01$). We ran a Tamhane's T2 Test comparing group means to identify specific differences. Tamhane's T2 Test does not assume equal variances among the samples.

Tamhane's T2 results indicate that the collection of learning tasks classified as *applying* was significantly different from the classifications of *remembering*, *understanding*, and *evaluating* ($p < 0.05$). *Applying* was not significantly different in number of queries per session from *analyzing* and *creating*. *Understanding* was also significantly different from *creating*, and *evaluating* was significantly different from *creating.* So, Hypothesis 1 is partially supported. By partially supported, we mean that at least one of the classifications were statistically different. All classifications statistically different from the other five would be a fully supported hypothesis. Figure 1 shows the mean queries per sessions of the six classifications.

Concerning topics per session, Using a one-way ANOVA, our results indicate that there is a significant difference among the groups ($F(5) = 8.613$, $p < 0.01$). Tamhane's T2 results again indicated significant differences among the classifications. *Applying* was significantly different from the classifications of *remembering*, *understanding*, and *evaluating* ($p < 0.05$). *Understanding* was significantly different from *creating*, and *evaluating* was significantly different from *creating*. Therefore, Hypothesis 2 is also partially supported. Figure 1 shows the mean topics per sessions of the six classifications.

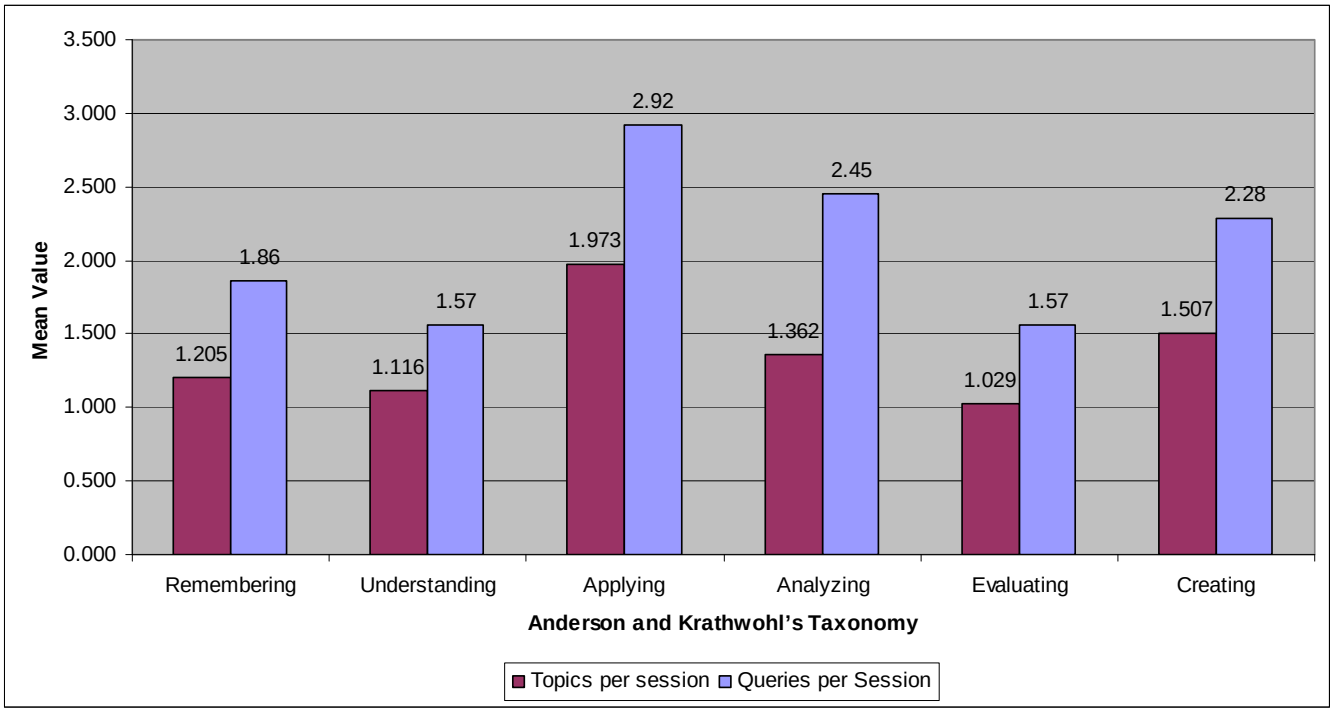


Figure 1. Queries and Topics per Session.

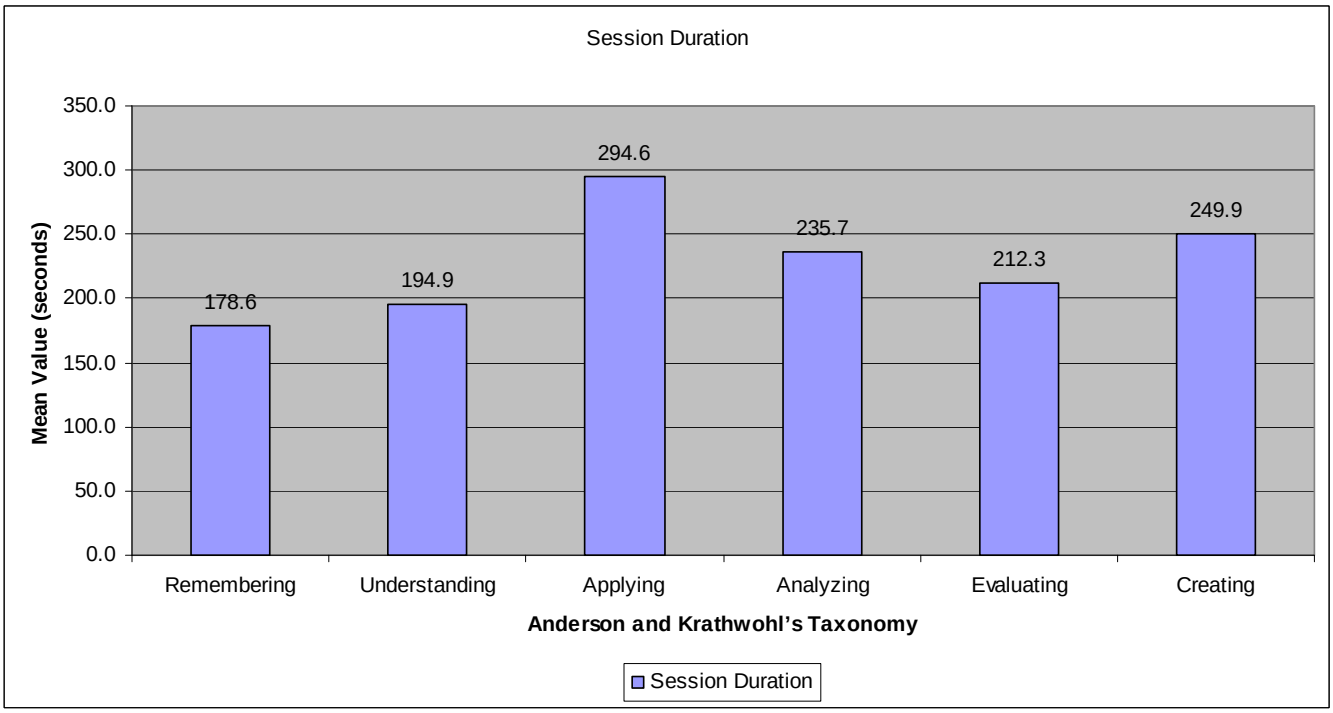


Figure 2. Session Duration (in Seconds)

For session duration, again using a one-way ANOVA, our results indicate that there is a significant difference among the groups ($F(5) = 2.68$, $p < 0.05$). Tamhane's T2 results indicate that the classification *applying* was significantly different from the classification of *remembering*. Hypothesis 3, therefore, is partially supported. Figure 2 shows the mean durations of sessions for each of the six classifications.

6. DISCUSSIOIN AND CONCLUSION

Research results indicate that learning appears to be an appropriate model through which to view searching. All

hypotheses were partially support using these designated searching characteristics.

Primarily, the middle classification of *applying* was generally statistically different than *remembering* and sometimes *understanding* (i.e., number of queries, number of topics, session duration, number of result pages viewed, and number of systems used). *Analyzing* was also statistically different from *remembering* (i.e., unique terms). Searching tasks at these learning levels appear to be the most challenging for searchers, exhibiting more complex searching characteristics. In some ways, one would expect these findings given that *remembering* and *understanding* are relatively 'lower level' cognitive tasks relative to *applying* and *analyzing.* However, in many cases *applying* and/or *analyzing* were also different from the 'higher level' cognitive tasks of *evaluating* and *creating* (i.e., number of queries, number of topics.

At the lower level of cognitive learning (*remembering* and *understanding*) and at the higher level (*evaluating* and *creating*), the exhibited searching characteristics are what one would deem indicative of relatively non-difficult searching tasks. At the lower levels, searchers seem to engage in fact checking and homepage-like finding activities. Interestingly, they seem to engage in the same activities at the higher level, presumably just to verify facts and information they already possess. While the higher levels tasks are more difficult, especially in terms of searching time, they appear to depend more on the users' creativity and viewpoints. The additional knowledge that searchers need to complete the task appear to be fact-finding tasks. Obviously, in these cases, searchers may be missing serendipitous findings and alternative viewpoints. This aspect would be a case for developing searching interfaces to facilitate exploratory searching. However, at the middle cognitive levels (*applying* and *analyzing*), the exhibited searching characteristics are characteristics of more complex searching needs.

The implications of this linkage between the cognitive processes, searching characteristics, and desired content are extremely beneficial for understanding the search process. Several researchers had lamented the lack of real system impact on information searching user studies, the shotgun approach [c.f., 11] to the identification of user characteristics, and the lack of granular searching models for the development of information searching systems. Marchionini [8] speaks of building supporting information tools if we can define kinds of information-searching, each with associated strategies and tactics. A learning model of information searching addresses all of these concerns.

What has been lacking is an inferential model that links the cognitive aspects of the user, searching characteristics, and type of content. From the results of this study, it appears that classifying information searching episodes by levels of the cognitive domain can possibly provide the linkage to content.

The findings of this research point to the designing of searching systems as learning systems. This would indicate features such as presenting a comprehensive set of results to the searcher, along with the most relevant results. It would also indicate that based on searching characteristics, one can infer user intent and content.

7. ACKNOWLEDGMENTS

The Air Force Office of Scientific Research and the National Science Foundation funded portions of this research.

8. REFERENCES

[1] Anderson, L. W. and Krathwohl, D. A., *A taxonomy for learning, teaching, and assessing: A revision of bloom's taxonomy of educational objectives*. New York: Longman, 2001.

[2] Bloom, B. S. and Krathwohl, D. R., *Taxonomy of educational objectives: The classification of educational goals, by a committee of college and university examiners. Handbook 1: Cognitive domain*. New York: Longmans, 1956.

[3] Croft, W. B. and Thompson, R. H., "I3: A new approach to the design of document retrieval systems," *Journal of the American Society for Information Science*, vol. 38, pp. 389-404, 1987.

[4] Donohew, L. and Tipton, L., "A conceptual model of information seek, avoiding, and processing," in *New models for mass communication research*, P. Clarke, Ed. Beverly Hills, CA: Sage, 1973, pp. 243-269.

[5] Halttunen, K., "Students' conceptions of information retrieval implications for the design of learning environments," *Library & Information Science Research*, vol. 25, pp. 307–332, 2003.

[6] Jansen, B. J., Booth, D., and Smith, B., "Using bloom's taxonomy of cognitive learning to model information searching," Under Review.

[7] Kuhlthau, C., "A principle of uncertainty for information seeking," *Journal of Documentation*, vol. 49, pp. 339-355, 1993.

[8] Marchionini, G., "Exploratory search: From finding to understanding," *Communication of the ACM*, vol. 49, pp. 41-47, 2006.

[9] Park, S., Bae, H., and Lee, J., "End user searching: A Web log analysis of NAVER, a korean Web search engine," *Library & Information Science Research*, vol. 27, pp. 203-221, 2005.

[10] Robertson, S., "Theories and models in information retrieval," *Journal of Documentation*, vol. 33, pp. 126-148, 1977.

[11] Saracevic, T., "Individual differences in organizing, searching and retrieving information," in *American Society for Information Science Annual Meeting*. vol. 28, 1991, pp. 82–86.

[12] Sperber, D. and Wilson, D., *Relevance: Communication and cognition*, Second ed. Oxford: Blackwell., 1995.

[13] Tang, R., "An integrated framework for Web searching research: Learning, problem solving, and search tasks," in *Fourth International Conference on Conceptions of Library and Information Science (CoLIS4)*, Seattle WA, 2002, pp. 49-67.

[14] van Rijsbergen, C. J., *Information retrieval*, 2 ed. London: Butterworths, 1975.

[15] Wang, P., Berry, M., and Yang, Y., "Mining longitudinal Web queries: Trends and patterns," *Journal of the American Society for Information Science and Technology*, vol. 54, pp. 743-758, 2003.

[16] Wolfram, D., "Term co-occurrence in Internet search engine queries: An analysis of the excite data set," *Canadian Journal of Information and Library Science*, vol. 24, pp. 12-33, 1999.

Focus on Results: Personal and Group Information Seeking Over Time

Gary Marchionini, Robert Capra, and Chirag Shah
School of Information and Library Science
University of North Carolina at Chapel Hill
100 Manning Hall
march@ils.unc.edu, rcapra3@unc.edu, chirag@unc.edu

1.INTRODUCTION

Information seeking is a fundamental human activity that is applied to an enormous range of information needs and exhibits diverse sets of individual behavioral nuances. Information needs range from fact retrieval to life-long interests in complex constructs and information-seeking behaviors range from brute force exhaustive search to sophisticated heuristics (e.g., building block, successive fraction, pearl growing, e.g., Hawkins & Wagers, 1982) and stochastic estimations. Today's search engines leverage content, links, metadata, and context such as time and place to return information based on searcher queries or selections. It is left to the information seeker to examine, interpret, and manage results independent of the search system, a condition that we aim to address here.

2.THE PROBLEM OF RESULTS

It is well known that people spend much more time examining results (both result sets and specific documents/pages) than composing queries (e.g., see Weinreich et al., 2007), however, the main emphasis of search engines is query processing, leaving the results examination to information seekers. Some search systems provide some results support. For example, Clusty (clusty.com) organizes results in clusters and the Cuil (cuil.com) provides spatial layouts of top-ranked search results. Coyle & Smyth (2007), Shneiderman and his colleagues (1994) and others have emphasized design of systems that support the entire search process and over the years, we have aimed to couple queries and results through highly interactive interfaces (e.g., the Relation Browser; Marchionini & Brunk, 2003; Capra & Marchionini, 2007). In this paper we focus on a framework for results management that will support searches over multiple sessions and possibly in collaboration.

Whereas most user-centered IR research focuses on query formulation and reformulation, we propose making the results of search the focal point of our work. By taking this novel approach to exploratory search, we aim to fill a gap between query-oriented IR and the personal/group information management systems (PIM and GIM, e.g., Erickson, 2006) that support information use. We propose a result space support system as a way to attack the multi-session exploratory and collaborative search problems and fill this gap in current research and development. To this end, we describe a result space architecture and outline one possible prototype based on this architecture that supports managing and optionally sharing result sets and items. Objects in the space include attributes such as search genesis (e.g., query), related objects (explicitly tagged, automatically linked), and temporal status (e.g., changes over time) and can be sharable individually or in aggregate.

Information seeking often takes place over multiple sessions. Current practices to deal with this include ad-hoc strategies. Email to self (Jones, et al. 2001; Whittaker, et al. 2006) has been documented as a particularly common strategy due in part to its ease of re-access from any location. Other strategies for re-access include bookmarks, saving and printing documents, and relying on being able to relocate information using search engine (Jones et al, 2001; Bruce et al. 2004; Aula et al. 2005). Studies have found that users struggle to make these ad-hoc strategies work for their needs and that personal information management is a challenge for users (Aula et al. 2005; Jones et al. 2001; Bruce et al., 2004). We aim to create a framework and tools for analyzing, saving, managing, and re-using results that will help overcome these ad-hoc strategies.

In the early days of online searching, professional intermediaries adopted techniques to reuse searches as they served many researchers with common interest (e.g., the Dialog search system allowed intermediaries to save sessions and query strategies more than 30 years ago). Komlodi's dissertation revealed the complexities of search history support in her study of searchers in law firms (Komlodi, 2002). She used participatory design to create prototype user interfaces that were in turn evaluated by legal searchers (Komlodi et al., 2007). She defined a search history framework with six primary components each with a hierarchical collection of factors: (scope of search history [21 factors at 3 levels], search context [28 factors at 4 levels], search history data [140 factors at 8 levels], search result management [24 factors at 4 levels], search history use [78 factors at 6 levels], and design features [80 factors at 5 levels].

Once relevant information is found, there are a variety of tools and services (e,g., RefWork, Zotero, Google Notebook, and Firefox Scrapbook) that support collecting and reorganizing search results. On-line tagging, bookmarking and social networking sites such as del.icio.us provide users with basic tools for storing bookmarks, tagging them, and sharing them with others. However, these systems treat the results as discrete and static objects and disassociate them from the queries that

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Conference'04, Month 1–2, 2004, City, State, Country.
Copyright 2004 ACM 1-58113-000-0/00/0004…$5.00.

generated them. We aim to close this gap and closely couple queries, result sets, and results (items) to facilitate reuse and ongoing information seeking over time.

Highly interactive search system styles such as dynamic queries (Alberg et al., 1992; Shneiderman, 1994) bridge the gap between discrete queries and result pairs to tightly couple queries and results and thus shift the focus from single-query retrieval to session-oriented retrieval. One of our primary goals in the work proposed here is to support tight coupling of inter-session searches. We moved in this direction with our Govstat project (find what you need, understand what you find, Marchionini et al.,2003) and aim to press further on this trajectory to more tightly couple queries and results over multiple sessions.

3.THE RESULTS FRAMEWORK

A general set of desiderata and vision for the system components follows. Such systems should allow people to easily:

1. Add results from new searches and result sets to their results space (perhaps coming from different sources and via automated processes);

2. Add annotations and tags to results, result sets, and queries;

3. Monitor changes to results, result sets, and queries over time;

4. Dynamically manipulate multiple result sets and queries to investigate overlaps, disjunctions, and changes over time; and

5. Selectively reuse and share results, result sets, and queries and their tags and annotations.

Figure 1 presents a schematic view of the Result Space, which consists of three dimensions: results, sessions, and users. A given cell in this cube consists of a set of Result Frames (RFs) for a result object (e.g., a web page, PDF file, video file) for a single session by a single user. Note that for queries that do not yield saved results, if the users wants nonetheless to save the query for future reuse, we will index a "no saved result" (null) entry that includes the query and other generative and contextual information. This will allow the user a more continuous history option within a single data model.

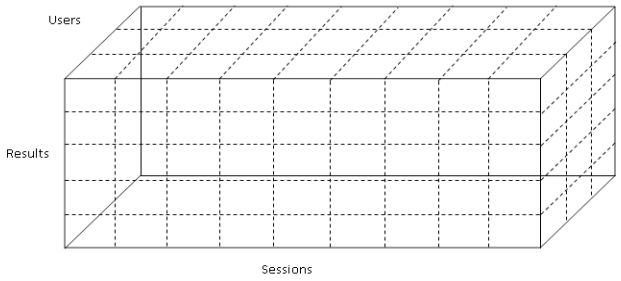


Figure 1. Results Space Model

A vertical stack of RFs represent a set of different results from a single session by a single user. An entire vertical slice represents the RFs for multiple users in a collaborative session (note that the session might be synchronous or not in practice, however, the figure suggests contemporaneous sessions). A horizontal (left to right in figure 1) stack of RFs represents a result that occurs in multiple sessions (e.g., revisiting or refinding, reusing the result in another session). The RFs are 'sets' because we want to explicitly store changes to a result that occur over time either by the user within a session (e.g., finding and saving the same result with different queries in a session) or more importantly, without user intermediation after a session (e.g., tagging or reuse by a collaborator, item revisions, or rank changes in the result set for the generating query). Thus, we treat results as dynamic objects within the result space that are active even when the user is not attentive to them and show these changes when the user is attentive. Considering RFs across the sessions and across the users supports the system's history mechanism and group information processing mechanism respectively.

A RF consists of a specific result (e.g., a web page, PDF file, video file) with various attributes and subspaces associated with it. Each RF has attributes: rank, tags, and notes. Rank stores the rank of the result in the query that generated it. Tags are provided by the user and/or other collaborative users in the group and are similar to the tags used in social network and bookmark system. Notes offer users a way to annotate results with personally meaningful contextual information. Unlike tags, notes allow free-form text that can include descriptions of the relevance of the result to the person or project, information for collaborative group members, a plan for using the result, or other annotations.

In addition to the attributes described above, each RF also contains the following subspaces: query, related, facets, world, and social. These subspaces can have attributes of their own. The query subspace contains attributes of the query that generated the result. These attributes are the query string, the source where the query was executed (e.g. Google, ACM Digital Library), and a listing of the top 100 results for the query (this number will be adjustable) from that source. The related subspace could contain items recommended by the source as being related to the result object such as explicit recommendations or outlinks.

Many result objects will have associated faceted metadata that can be stored in a facet subspace. For instance, video objects might have metadata facets for duration, genre, creation date, and source. Facets consist of pairs of a facet name (e.g. "size") and a counterpart value (e.g. "Medium"). Faceted metadata may come from the data source itself, may be obtained from shared collaborative information, or may be automatically generated by classification engines. Our research group has extensive experience building interfaces and classification engines to support faceted search (Capra et al., 2007; Zhang and Marchionini, 2005; Efron, et al., 2004; Marchionini and Brunk, 2003).

A world subspace represents external attributes that are obtained from sources other than the source where the information was located. In most cases, the other sources will be Web-based services. For instance, if the result is a blog entry, we may consult a web search engine to obtain information about the PageRank of the blog page and the inlinks to that blog.

Collaborative contributions about the result are captured in the social subspace. The community around the user may be a close work group or a broader social network. For instance, if the result is a book, the social subspace might record how other people have

rated the book and include all the associated reviews from a social network. A permissions mechanism will be implemented at the RF, subspace, and attribute levels so that users can selectively share at fine grains. Each of these attributes and subspaces of results can be used to organize, filter, manage, and re-use result spaces.

4.PROTOTYPE IDEAS

In order to support these user activities, we outline ideas for a Results Space (RS) system. Such a system must manage shared access at different granularities (allow a user to specify which items and attributes are shared with whom) and automatically update contextual information over time. Information will be gathered from client devices (upon user initiation) as the user engages in web browsing and searching. To support the ability to start an information seeking session on one device and continue it on another device or in later session, information gathered during the search will be stored on a server. We intend to build the client and server software on top of established open-source software components and provide simple installers so that users could easily use the RS system on a local server or intranet.

Storing queries, result sets, and annotations on a central server has a number of privacy issues for many users and organizations. Emerging bookmark and web clipping notebook services (e.g. del.icio.us and Google notebook) require users to set up accounts and store information on their central servers. An alternative approach taken in the UCAIR project (Shen et al., 2005a, 2005b) is to store search history on the client machine, however, this does not allow people to use multiple platforms for their ongoing work. There is a classic tradeoff between supporting remote access (storing information on a server) and providing more privacy and security (store information in only one place – on the user's PC). One compromise that many companies use is to host their own servers (e.g. corporate email servers) to provide remote access while gaining a level of control of the privacy and security for their organization. In fact, this type of intra-net level use is one of the situations where we envision the collaborative aspects of the RS system being most useful –knowledge workers on a project team within an organization collaboratively conducting searches and synthesizing results in order to create their work products. The RS system outlined here must provide the tools and central coordination needed to support such collaborative research work.

One of the interfaces we anticipate including in the RS system is a web browser toolbar that allows users to interact with current search results as well as with previous result sets. Toolbars are a commonly used, unobtrusive interface that can provide a lightweight way to add and annotate results while also providing controls that can expand or use the main web browsing space to support additional interactions such as visualizations.

In the RS system, the individual result items found by the user as part of their information seeking will be the main focal objects. However, individual results, result sets, and queries will all be first-class objects in the proposed architecture. This means that they can all be stored, manipulated, composed, and inspected as part of the system. This style of architecture will allow different controls, visualizations, and operations to be easily developed and "plugged-in" to the RS system. We have experience developing complex query and result set models using this style of architecture from our prior work on the Relation Browser (Capra and Marchionini, 2007a, 2007b) and Context Miner (Shah & Marchionini, 2007) systems. In the work proposed here, we will extend the architecture and interfaces to support: 1) multiple sessions (extending the model across time), 2) multiple devices (extending the model to support full and limited feature sets), 3) annotations and connections to results, result sets, and queries, and 4) collaborative views and reuse of the data.

5.ACKNOWLEDGMENTS

This work was supported by NSF grant IIS 0812363.

6.REFERENCES

[1] Ahlberg, C., Williamson, C. and Shneiderman, B. (1992). Dynamic queries for information exploration: An implementation and evaluation. In Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems, pp. 619-626.

[2] Aula, A., Jhaveri, N., Käki, M. (2005). Information Search and Re-access Strategies of Experienced Web Users, In Proc. of WWW 2005, Chiba Japan, 583-592.

[3] Bruce, H., Jones, W., and Dumais, S. (2004). Keeping and re-finding information on the web: What do people do and what do they need? In Proc. ASIST 2004, Chicago, IL, Information Today, Inc., October, 2004.

[4] Capra, R. and Marchionini, M. (2007). Faceted Browsing, Dynamic Interfaces, and Exploratory Search: Experiences and Challenges. Proceedings of the Workshop on Human-Computer Interaction and Information Retrieval (Cambridge, MA, October 23, 2007), 7-9.

[5] Capra, R., Marchionini, G., Oh, J. S., Stutzman, F., and Zhang, Y. (2007). Effects of structure and interaction style on distinct search tasks. Proceedings of the 2007 Conference on Digital Libraries (Vancouver, BC, Canada, June 18 - 23, 2007).

[6] Coyle, M. & Smyth, B. (2007). Supporting intelligent web search. ACM Transactions on Internet Technology. 7(4), 1-31.

[7] Efron, M., Elsas, J., Marchionini, G., and Zhang J. (2004). Machine learning for information architecture in a large governmental website. Proceedings of the 4th ACM/IEEE Joint Conference on Digital Libraries (Tuscon, AZ, June 7-11, 2004), 151-159.

[8] Erickson, T. (2006). From PIM to GIM: personal information management in group contexts. Commications of the ACM, 49, 1 (Jan. 2006), 74-75.

[9] Hawkins, D. & Wagers, R. (1982). Online bibliographic search strategy development. Online, 6(3), 12-19

[10] Jones, W., Bruce, H., & Dumais, S. (2001). Keeping found things found on the Web. Proceedings of the 2001 ACM CIKM 10th International Conference on Information and Knowledge Management (Atlanta, GA, November 5-10, 2001) New York: Association for Computing Machinery, 119-134.

[11] Komlodi, A. (2002). Search history for user support in information-seeking interfaces. Unpublished doctoral dissertation, University of Maryland, College Park.

[12] Komlodi, A., Marchionini, G., and Soergel, D. (2007). Search history support for finding and using information: user interface design recommendations from a user study. Inf. Process. Manage. 43, 1 (Jan. 2007), 10-29.

[13] Marchionini, G., Haas, S., Plaisant, C., Shneiderman, B., & Hert, C. (2003). Toward a statistical knowledge network. Proceedings of the Third Conference on Digital Government Research dg.03 (Boston, May 18-21, 2003). P. 27-32.

[14] Marchionini, G. and Brunk, B. (2003). Towards a general relation browser: A GUI for information architects. Journal of Digital Information, 4(1).

[15] Shah, C. & Marchionini, G. (2007). Preserving 2008 US Presidential Election Videos. Paper at the 7th International Workshop on Web Archiving and Digital Preservation (IWAW'07). http://www.ils.unc.edu/vidarch/Shah-IWAW07.pdf

[16] Shen, X., Tan, B., & Zhai, C. (2005a).UCAIR: Capturing and exploiting context for personalized search. In Proceedings of 2005 ACM Conference on Research and Development on Information Retrieval - Information Retrieval in Context Workshop (IRiX'2005).

[17] Shen, X., Tan, B., & Zhai, C. (2005b). Context-sensitive information retrieval using implicit feedback. In Proceedings of 2005 ACM Conference on Research and Development on Information Retrieval (SIGIR '05), 43-50.

[18] Shneiderman, B. (1994). Dynamic Queries for Visual Information Seeking, IEEE Software, 11(6): 70-77.

[19] Weinreich, H., Obendorf, H., Herder, E., & Mayer, M. (2008). Not quite the average: An empirical study of Web use. ACM Transactions on the Web, 2(1), Article #5.

[20] Whittaker, S.; Bellotti, V.; Gwizdka, J. (2006). Email as personal information management. Communications of the ACM. 2006 January; 49 (1): 68-73.

[21] Zhang, J. and Marchionini, G. (2005). Evaluation and evolution of a browse and search interface: relation browser. In Proceedings of the 2005 National Conference on Digital Government Research (Atlanta, Georgia, May 15 - 18, 2005). ACM International Conference Proceeding Series, vol. 89. Digital Government Research Center, 179-188.

SocialRank: An Ego- and Time-centric Workflow for Relationship Identification

Jaime Montemayor, Chris Diehl, Mike Pekala, David Patrone
Milton Eisenhower Research Center, The Johns Hopkins University Applied Physics Laboratory
jaime.montemayor@jhuapl.edu

ABSTRACT

From instant messaging and email to wikis and blogs, millions of individuals are generating content that reflects their relationships with others in the world. Since communication artifacts are recordings of life events, we can gain insights into the social structure, attributes, and dynamics from this communication history. To help an analyst explore, discover and identify important social structures in these online communication archives, we have developed SocialRank, an ego- and time-centric workflow for identifying social relationships in an email corpus. This workflow includes four high-level tasks: discovery, validation, annotation and dissemination. Given the volume of data and complex relationship structures that confront the analyst, an effective analytic process must dramatically accelerate the discovery of relevant relationships, facilitate the recordings of assertions and validations of these discoveries, and produce reports for the dissemination of an analyst's findings. SocialRank supports these tasks, through the integration of relationship ranking algorithms with timeline, social network diagram, and multidimensional scaling visualization techniques.

Categories and Subject Descriptors

H.5.2 [**Information Systems**]: Information Interfaces and Presentation—*User interfaces*; H.4.3 [**Information Systems**]: Communications Applications—*Information browsers*; I.3.6 [**Computing Methodologies**]: Methodology and Techniques—*Interaction techniques*

Keywords

Information Visualization, Visual Analytics, Machine Learning, Retrospective Analysis, Multidimensional Scaling

1. INTRODUCTION

Millions of individuals are generating digital content that reflects their (online and offline) relationships with others in the world. As groups and organizations increasingly leverage online means of communication and collaboration, there is an opportunity for analysts to develop new insights regarding the structure, attributes, and dynamics of the underlying social network.

In intelligence analysis and litigation support, analysts construct validated social networks from communication events. During this process, there are two distinct tasks: entity resolution and relationship identification. While we recognize that a process needs to support both tasks to be successful, in this paper we limit our discussion to the relationship identification task[1].

In the sections that follow, we first highlight the algorithmic components that support discovery and validation. Then we describe how these components are integrated in SocialRank with visualization and interaction methods to facilitate annotation and dissemination, thus completing the analytic workflow.

2. RELATIONSHIP IDENTIFICATION

Informal, online communications are composed of structured and unstructured data. At the most basic level, this includes the network references corresponding to the sender and one or more recipients, the date and time of the communication and the message content. We define a communications archive as a set of observed messages exchanged among a set of network references. Every archive has a corresponding communications graph that represents the message data as a set of dyadic (pairwise) communication relationships among the network references. The task of relationship identification involves identifying a mapping from the dyadic communication relationships to one or more social relationships of interest. In this section, we discuss two classes of algorithms that support the analyst in the construction of the underlying social network: content-based and activity-based relationship ranking.

2.1 Content-Based Relationship and Message Ranking

We envision that an analyst navigates a communications graph by following and incrementally investigating ego networks. We use a two-step process to identify relevant social relationships (e.g. manager-subordinate) within a given ego network. Using a scoring function learned from message content associated with labeled ego networks [2], communica-

[1]Entity resolution refers to the mapping of network references to their corresponding entities (e.g., [1]). Relationship identification refers to the identification of relevant communications that are indicative of a given relationship type.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
HCIR 2008 Seattle, Washington USA
Copyright 200X ACM X-XXXXX-XX-X/XX/XX ...$5.00.

tion relationships are first ranked according to their relative likelihood of exhibiting a specified social relation; then, the messages within each communication relationship are ranked according to their relative support for the relationship rank.

2.2 Activity-Based Relationship Ranking

Once we have identified a particular social relation of interest, we often want to discover other communication relationships that may indicate the existence of group structure within which the identified social relationship is embedded. We achieve this by comparing the patterns of communication between a given reference communication relationship and the remaining relationships within the ego network. This provides a purely structural approach that helps the analyst establish relationship similarity, independent of content, thereby complementing the content-based rankers learned from analyst annotations.

Given a collection of activity vectors that represent the temporal rhythms of the relationships in the ego network, we use metric multidimensional scaling to generate a two-dimensional configuration of points that represents the relative similarities of the relationships, as captured by the Euclidean distance among the original activity vectors in the high-dimensional vector space. By selecting a particular communication relationship to serve as the reference, the remaining relationships can be resorted based on their distance from the reference.

3. SOCIALRANK

The utility of a workflow for relationship identification is dependent on its ability to 1) dramatically accelerate the discovery of relevant relationships, 2) validate and track hypothesized relationships, and 3) generate reports of an analyst's findings. SocialRank [3] facilitates discovery and validation through a combination of ranking algorithms and information visualization techniques. An analyst discovers interesting relationships using the timeline (Figure 1), multi-dimensional scaling (MDS), network structure (Figure 2), network evolution (Figure 3), and message viewers (right panels seen in Figures 1 and 2).

The timeline viewer displays an ego's pairwise communication relationships over time[2]. The content-based relationship ranker orders the communication relationships in terms of their relative likelihood of exhibiting a user-specified social relationship (e.g. manager-subordinate). In order to assert that such a social relationship exists between an ego and alter, the analyst inspects the communication relationship timelines of the candidate alters. Since the most important messages supporting the relationship are indicated with visual cues on the timeline, instead of wading through hundreds of email messages, the analyst is directed to a few messages to read in detail to assess whether the content supports the relationship. Hence, this combination of relationship ranking and visualization can accelerate the discovery of messages containing supporting evidence.

When a message supports a social relationship, an analyst asserts this claim and creates an annotation. SocialRank then automatically inserts the new validated relationship into the network structure diagram (Figure 1), and remembers the corresponding email message and notes (Figure 2).

[2]In SocialRank, an egocentric analysis is an examination of the relationships between a focal actor (individual), called an ego, and other actors, called alters.

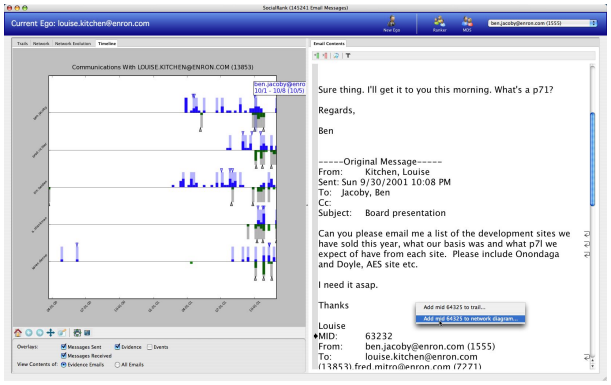


Figure 1: The timeline viewer displays an ego's pairwise communication relationships on a timeline. The relationship ranker identifies (light shading and triangles) the time intervals that contain messages that likely express this relationship. After reading a message, if an analyst is satisfied that the content suggests a social relationship exists between the ego and alter, she can immediately create an annotated relationship and assign the message as the validating evidence.

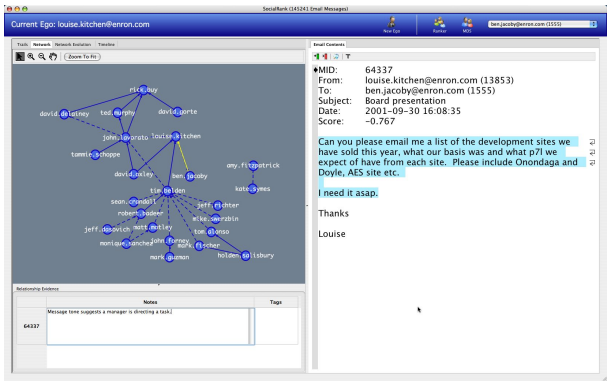


Figure 2: SocialRank automatically tracks an analyst's discoveries about social relationships and their corroborating email messages and annotations.

The MDS diagram complements the timeline. It relies on a structural comparison between the reference and candidate communication relationship over a specified time interval. Thus, once a reference ego-alter pair has been identified with a social relationship, an analyst can use the MDS diagram to reveal additional candidates by examining other communication relationships that exhibit similar patterns relative to the reference.

The network diagram in Figure 2 represents the captured knowledge of social relationships, their corresponding validating messages and the analyst's annotations. This static diagram cannot represent relationship dynamics. We developed the network evolution viewer to incorporate the temporal attribute (Figure 3). In this diagram, SocialRank tracks the evolution of a social network (centered on an ego) and shows the temporal locations of the messages (evidence) that support the relationship.

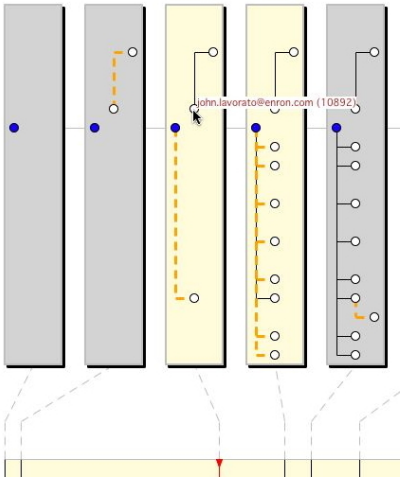


Figure 3: The egocentric network evolution viewer shows an analyst's understanding of an egocentric organizational structure and the temporal locations of the supporting evidence.

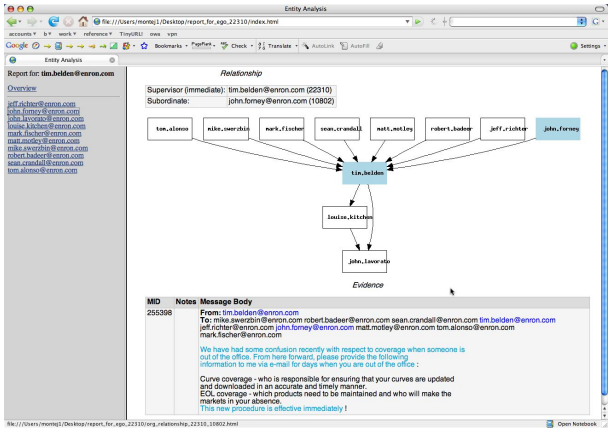


Figure 4: This HTML page shows the message evidence that supports an asserted social relationship.

Finally, SocialRank completes the analytic workflow by supporting the dissemination phase of the process, which includes three components: collection, ordering, and reporting (Figure 4). When an analyst is ready to present the results of her work, SocialRank collects the email evidence and user annotations about the entities who are connected by a social relationship to an ego. Next, the analyst can reorder these elements to facilitate a compelling narrative. Finally, SocialRank generates an HTML-based report, including a egocentric network diagram summary, followed by the evidence and comments that validate each relationship in that network.

4. NEXT STEPS

Our next machine learning objective is to develop and integrate an incremental learning capability into SocialRank so that rankers can be incrementally trained as the analyst provides annotations during exploration of the communications archive. To move toward automated incremental learning, a series of additional challenges must be addressed such as learning from partially labeled ego networks with uncertainty in the time extent of the social relationship and automated model and feature selection. Such methods will be integrated with new information visualization techniques to better represent time in both the network evolution and MDS views.

5. ACKNOWLEDGMENTS

This work was supported by an internal research and development grant from JHU/APL.

6. REFERENCES

[1] M. Bilgic, L. Licamele, L. Getoor, and B. Shneiderman. D-dupe: An interactive tool for entity resolution in social networks. In *Visual Analytics Science and Technology (VAST)*, Baltimore, October 2006.

[2] C. Diehl, G. M. Namata, and L. Getoor. Relationship identification for social network discovery. In *AAAI '07: Proceedings of the 22nd National Conference on Artificial Intelligence*, July 2007.

[3] J. Montemayor, C. Diehl, M. Pekala, and D. Patrone. Socialrank: An ego- and time-centric workflow for relationship identification. In *VAST Posters*, 2008.

SketchBrain: An Interactive Information Seeking Interface for Exploratory Search

Hogun Park, Sung Hyon Myaeng, Gwan Jang, Jong-wook Choi, Sooran Jo, Hyung-chul Roh
School of Engineering, Information & Communications University (ICU)
119, Munjiro, Yuseong-gu, Daejeon, 305-732, Korea
+42-866-6210, 82

{gsgphg, myaeng, ilys23, pudidic, ddangly-, nuunmir }@icu.ac.kr

ABSTRACT
As the Web has become a commodity, it is used for a variety of purposes and tasks that may require a great deal of cognitive efforts. However, most search engines developed for the Web provide users with only searching and browsing capabilities, leaving all the burdens of manipulating information objects to the users. In this paper, we focus on an exploratory search task and propose an underlying framework for human-Web interactions. Based on the framework, we designed and implemented a new information seeking interface that helps users to relieve cognitive burden. The new human-Web interface provides a personal workspace that can be created and manipulated cooperatively with the system, which helps the user conceptualize his information seeking tasks and record their trails for future uses. This interaction tool has been tested for its efficacy as an aid for exploratory search.

Categories and Subject Descriptors
H.3.7 [Information Storage and Retrieval]: Information seeking Interface

General Terms
Documentation, Design, Experimentation.

Keywords
Information Seeking Interface, Exploratory Search

1. INTRODUCTION
For a traditional Web search engine, the process of querying and viewing the results is usually regarded as a single, isolated session that ends in itself. As the Web has become a commodity, however, it is used for a variety of tasks in many different ways, encouraging new paradigms in information seeking (e.g. berrypicking [1], information foraging [2], and sense-making [3]). However, most popular commercial search engines have taken a conservative position and adhered to the traditional model, leaving all the rest of the information seeking and related tasks to the user. More specifically, the user has all the burdens of manipulating the information objects that have come to his attention in a series of search activities.

An area in which this type of cognitive burden affects significantly is exploratory search. An exploratory search task [4][5] is to investigate on the background information of a topic or gather information sufficient to make an informed decision. For example, assume that a user is considering purchasing a DMB (digital multimedia broadcasting) receiver. The user would want to learn more about the DMB technology and the manufacturers of various products related to it, so that he can select the provider and the products that best suit the needs. We believe that most existing search engines and their interfaces are not satisfactory for exploratory tasks, because of the following.

First, compared to the task of searching for specific or known items, an exploratory search task usually requires users to send a series of queries during a search session, visit more new domains, and revisit previously visited sites (especially branch pages) [5]. These activities together mean a significant amount of information and workload that traditional search engines have rarely attempted to reduce. The workload is associated with representing information needs [14], determining informativeness [15], and memorizing previously explored information [16]. Without explicit support from a search engine, the difficulties resulting from the workload are left as a cognitive burden to the user. Second, there are narrow interaction channels for incorporating user interests. In an exploratory search, a user needs to build up background information on a topic gradually until she feels that a sufficient amount of information has been gathered for the given task. As such, it is important to incorporate the users' interest and the information that has been found as the system processes the current query. However, current search systems rarely support the notion of "session" and interactions explicitly. While the one-time query/result model is simple and natural with HTTP, it ignores what has been done by the user in her attempt to change her anomalous state of knowledge [17]. Although there have been some attempts to infer user interest explicitly [7][8][9], implicitly [18], or both [19], the problem remains challenging, especially within the context of user-system interactions.

Given the limitations of traditional search engines for an open-ended, exploratory search task, we propose a new interaction tool that can provide an interface between a user and a search engine, called *sketchBrain*. Our aim is to provide an effective interaction

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
HCIR'08, Oct 23, 2008, Redmond, WA.
Copyright 2008 ACM 1-58113-000-0/00/0004…$5.00.

environment that facilitates the series of activities in an exploratory search of the Web.

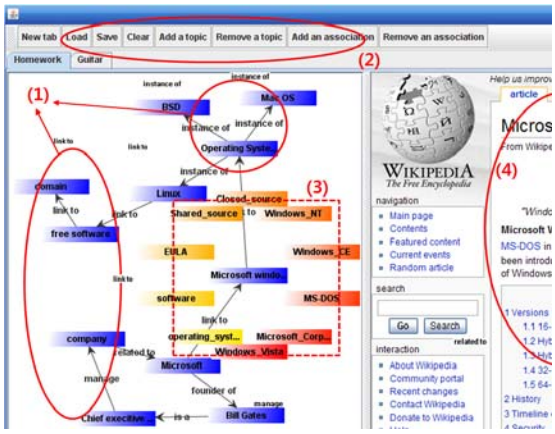


Figure 1. An example screen shot of *sketchBrain*

There are several noble features in this interaction environment. First of all, *sketchBrain* keeps track of query trails and post-query navigation trails (based on the click stream following the issued queries) and allows the users to conceptualize them. For an information seeking activity, a trail is sketched on the user's workspace of *sketchBrain*. Over the trail, the user can associate user-defined topics and system-provided semantic associations between topics using the annotation facility in *sketchBrain*. The annotation over trails means cognitive structure or the explication of user's conceptual view of the information objects being explored through interactions with the Web. It represents users' information need and affects next cognitive behaviors, so it plays an important role of reducing cognitive burden. Moreover, it has a potential for making personal metadata that can be shared with others and improving searching/browsing capability. In essence, the workspace serves as a rich memory for the past and current search efforts, which can be accessed later.

Second, our interaction tool is equipped with operations on the objects created and manipulated in the workspace. In addition to the annotation facility, *sketchBrain* allows users to manipulate the objects for their information seeking tasks. Implicit operations such as project, select, and classification (to be described in Section 3) can be utilized for the activities necessary for an exploratory search.

Third, *sketchBrain* has an intelligent path recommendation algorithm that can help users choose the most promising page to be explored at the next step in navigation. It assists users in determining informativeness of the pages that can be explored at the next step quickly.

A screenshot containing the user interface of *sketchBrain* is shown in Fig. 1. On the left is the user workspace where three workflows are sketched as indicated by (1). Using this tool, click-through data can be recorded as much as the user wishes to remember for future use. For example, whenever a user visits a new page, a new node is created and connected to an originated page or a query with a directed edge. They can be modified by manipulation tools (2), and, via this manipulation and the workspace, the user represents own conceptual understanding. In addition to this feature, our system can provide the relevant context of a specific page (like the one pointed by (4)) through time-variant multiple spreading activations (3), which can be used as a guidance for further navigation. The degree of relevance is determined by the algorithm and is shown in various colours (red indicates the most relevant one).

The remainder of this paper is composed of underlying model (Section 3), the interaction framework for supporting an exploratory search task (Section 4), and empirical evaluation via user studies (Section 5.)

2. RELATED WORK

Various information seeking interfaces have been proposed to support complex information seeking activities. Sketchtrieve [6] employs Cognitive Dimension Framework to map out the design space and provides an unstructured canvas. In this canvas, searchers can freely represent queries and corresponding search results with an intuitive interface by using typographic and layout cues that lie outside of a formal notation. Buchanan et al. [7] introduces information seeking workspace called Garnet. They exploit implicit knowledge that can be discovered from the contents in the workspace and try to find direct connections between the workspace and digital libraries. They utilize spatial parsing to extract profiles of documents and use them to learn a lexical classifier. This classifier is to identify newly searched documents that are relevant to each parsed cluster. Martin and Jose [8] suggest a personal information retrieval tool that employs a folder-like structure, so that searchers can bundle search results into folders. In addition to the interface that searchers can freely organize results, it assists query formulation and recommends hot relevant documents to each folder. Harper and Kelly [9] employ a topical structure for relevance feedback. Their interface allows users to save documents in user-defined piles for similar documents, which could be used for relevance feedback. These approaches suggest new information seeking environments with some assistance. However, their design goals are not to support exploratory search explicitly, and the systems were not tested as such. Our interface provides users with a cooperative workspace and a proactive assistance, explicitly aiming at exploratory searching tasks.

3. THE UNDERLYING MODEL

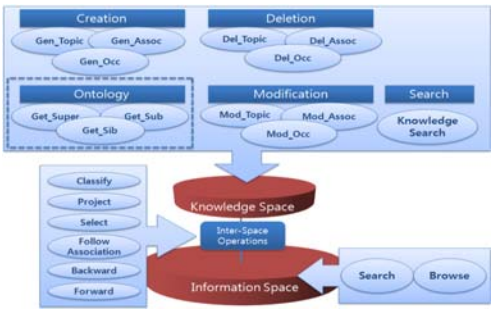


Figure 2. A conceptual view of the two-level model

Our interaction tool and user interface are based on our two-level model that explicates information and knowledge spaces where user information seeking activities take place. Fig. 2 depicts a conceptual view of the underlying model and the relationship between the information and knowledge spaces and the operations.

We attempt to separate users' conceptual work space into two levels and define operations on each space and inter-space operations [see [10] for details]. The set of operations in Fig. 2 is by no means complete, and we intend to expand it as additional needs arise.

4. INTERACTION FRAMEWORK

We have designed an interaction framework and implemented a prototype system, called *sketchBrain* that includes a search engine and the interaction tool, capturing the key ideas of the two-level model described before. *sketchBrain* is implemented with an open source graphics library (http://www.jgraph.com/) in Java, which we extended for our purposes.

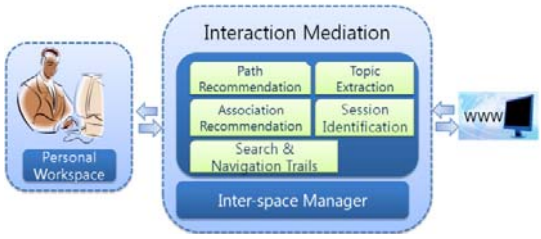


Figure 3. *sketchBrain* interaction framework

As in Fig. 3, the framework connects users with the Web through Interaction Mediation. While a user has a virtual workspace, the Web side is assumed to have a conventional search engine and browsing facilities. When the user searches/navigates the Web and attempts to make informed decisions based on the information found, Interaction Mediation provides a support with the goal of relieving his cognitive burden in the information seeking process. It consists of various tools that facilitate users' information seeking activities in terms of searching and browsing and work space creation/manipulation. Modules for topic extraction, association recommendation, and searching&browsing trails tracking assist cooperatively to construct personal knowledge structure on workspace. Path recommendation and session identification help to facilitate interaction between users and digital library. Inter-space Manager associates the personal cognitive structure with raw information in WWW and provides facilities to manipulate them. For further motivation and details of *sketchBrain*, please refer to [11].

5. EXPERIMENT

In the first experiment, we tested whether the proposed tool helps reducing users' workload (i.e. cognitive burdens) in exploratory search, the primary motivation for devising the proposed method. In the second experiment, we tested the tool for its usefulness in reusing previously encountered information. More specifically, it tested how the proposed tool helps users in performing tasks that require organizing and remembering the results from searching and browsing.

Experiment 1: Reducing Workload

Our first interest was to find out whether the system implemented based on the two-level model would help reducing workload of users. Given the motivations of our work, workload is a reasonable measurement to test the tool's efficacy because it measures how much effort is required to complete an exploratory search task. In this experiment, we used a special instrument, subjective workload assessment technique (SWAT) [12]. This method has been utilized for evaluating three criteria: time, mental effort, and stress.

We asked the participants to perform a total of 10 exploratory search tasks in the Wikipedia environment where the articles were judged for usefulness in learning background and detailed information for exploratory search tasks. In this experiment, we utilized a simple English Wikipedia, and evaluated efficacy of our information seeking interface as an aid to exploratory search. Each task has one topic selected from the topics of 10 different Wikipedia categories. For a more realistic exploratory search environment, we provided blank forms that they had to fill out. The forms are composed of two parts: semantic annotation and summarizing. Semantic annotation is to annotate information about what related entities appear in texts, and summarization means answering non-factoid questions like "writing a state of the art" and "writing important background information". To minimize potential biases like leaning effects, the participants applied two methods, with and without the interface, in an alternating fashion.

Table 1. The result of SWAT

	with interface (Average SD)	without interface (Average SD)	Difference
Time	1.6 (0.55)	1.8 (0.45)	+ 0.2
Mental effort	1.2 (0.45)	2.4 (0.55)	- 1.2
Stress	1.8 (0.45)	2.2 (0.45)	- 0.4
Total	4.6 (0.89)	6.4 (0.89)	- 1.8

The participants' rates of SWAT range between 1 (the best) and 3, and the result of workload analysis is presented in Table 1. Our interface received a mean score of 4.6, which is a significant improvement over the case without the interface. In particular, the difference was the greatest for mental efforts as intended and expected for the interface. These observations showed that our new information seeking interface helped reducing workload in three different ways in the task of exploratory search.

Experiment 2: Information Reuse

Since our two-level model and its manifestation as a tool were devised to help users reducing cognitive efforts in information seeking processes, manifested by searching and browsing activities, we decided to focus on information reuse activities in information seeking. In the web environment, users often have to skim through an overwhelming amount of information, suffering from information overload, before their goals are achieved. In this experiment, The three methods, the Favorites tool, SIS [13], and *sketchBrain*, were compared in six different tasks by ten groups of users, each consisting of three undergraduate students. In total, 30 users were employed for six different tasks using three different methods. The six tasks consist of questions in six different domains like Medicine and Sports. The tasks were designed as follows. For a task, the participants (users) were first asked to read 30 pre-selected web pages. One minute per page was given to simulate an information skimming situation. The participants were then asked to organize the pages using the given tool within one minute. After the preparation stage, they were given three information hunting questions elicited from the 30 pages they read. The participants were timed for completion of each question answering. Since the maximum time given to each question was five minutes, the time taken for an unsolved question was assumed to be solved in five minutes, the maximum. In order to minimize user dependency and learning effects, the users were

assigned to six tasks using three different methods in an alternating fashion. Each user evaluated each method twice for different tasks, and each task was given to the three users in an effort to minimize user dependency. Three users used the three methods in different sequences for different tasks so that there is little learning effect on average.

To ensure that every participant has some familiarity with the three tools, we gave them a tutorial with 10 minutes of practice sessions in the same place with all the participants together.

Table 2. ANOVA result

			95% Confidence Interval for Mean	
Methods	Mean	Std.Deviation	Lower Bound	Upper Bound
1: Favorites	87.69	98.82	62.16	113.22
2: SIS	70.09	67.67	52.61	87.57
3: Our Tool	50.33	43.78	39.02	61.64

The comparison result is shown in Table. 2. It took about 50 seconds on average to solve the problems using our tool, but 88 (about 76% longer) and 70 seconds (about 40% longer) using the Favorites tool and SIS, respectively. Although SIS didn't require any extra user efforts to organize the pages, the time spent on the organization was only one minute, once for all the tasks. If the initial investment for our tool is spread across all the questions, the extra time spent is very small. In ANOVA analysis, it shows that the mean for our tool was better than those of Favorite and SIS. ANOVA puts all the data into one number (F) and gives us one P for the null hypothesis. The value was equal to $F(2,177)=3.866$ ($p < 0.05$), and the difference was reliable at the 95% confidence level. It means that users were more likely to say that our tool had superior information reusability.

6. CONCLUSION

We have proposed a new information seeking interface for extracting/utilizing cognitive personal knowledge structure, which explicates operations at the knowledge level and across the information and knowledge spaces in addition to the typical information level operations, searching and browsing. The tool we developed, which is a limited manifestation of the model, was first tested how the tool is helpful to reduce cognitive burden. Based on the encouraging results, we conducted a more focused and carefully designed experiment to evaluate the tool's utility in reusing a relatively large amount of information that has been encountered. In comparison, our tool was superior to the others in supporting information reuse tasks. The result indicates that our novel approach, the two level model and the associated operations, is very promising and worth further study. First of all, the two-level model can be extended further and implemented in other ways with different emphases. For example, it would be useful to search using a topic-association-topic triplet as a query. In this case, information objects need to be indexed accordingly. Second, automatic generation of topics and associations require further research, which is essential to reducing users' burden in constructing their own knowledge space. Finally, a complete system based on the two-level model must be deployed to a real user environment for more extensive experiments.

7. REFERENCES

[1] Bates, M. 1989. The design of browsing and berry picking techniques for the online search interface. Online Re-view. 13, 5, 407-431.

[2] Pirolli, P. & Card, S.K. 1995. Information foraging. *Psychological Review*, 106, 643-6753.

[3] Russell, D.M. et al. 1993. The cost structure of sensemaking, Proc. of CHI 1993. 269-276.

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

[5] White, R. W. and Drucker, S. M. 2007: Investigating behavioral variability in web search. Proc. of WWW 2007. 21-30.

[6] Hendry, D. G., and Harper, D. J. 1997. An informal information-seeking environment. Journal of the American Society for Information Science. 48, 11, 1036-1048.

[7] Buchanan, G., et al. 2002. Spatial Hypertext as a Reader Tool in Digital Libraries. LNCS 2539, 13-24.

[8] Martin, I., and Jose, J.M. 2004. Fetch: A personalized information retrieval tool. Proc. of RIAO'2004 Con.

[9] Harper, D. J. and Kelly, D. 2006. Contextual relevance feedback. Proc. of IIiX 2006, 129-137.

[10] Park, H., et al. 2007. Personalized Knowledge Structure for Exploratory Search and Information Reuse, http://ir.icu.ac.kr/~phg/snb_techReport1.pdf.

[11] Park, H., et al. 2008. Interactive Information Seeking Interface for Exploratory Search. Proc. of ICEIS 2008. 276-285

[12] Reid, G.B. and Nygren., T.E. 1988. The subjective workload assessment technique: A scaling procedure for measuring mental workload. P.A. HANCOCK and N. MESHKATI, eds. Human Mental Workload. Amsterdam: North Holland.

[13] Dumais, S., et al. 2003. Stuff I've seen: a system for personal information retrieval and re-use. Proc. of SIGIR 2003, 72-79.

[14] Taylor, R. 1968. Question-Negotiation and Information Seeking in Libraries. College & Research Libraries, 29, 3, 178-194.

[15] Teevan, J., et al. 2004. The perfect search engine is not enough: a study of orienteering behavior in directed search. Proc. of CHI 2004, 415-422.

[16] McKenzie, B. and Cockburn, A. 2001. An empirical analysis of web page revisitation. Proc. of HICSS 2001, CD Rom.

[17] Belkin, M.J. 1980. Anomalous states of knowledge as a basis for information retrieval, Canadian Journal of Information Science. 5, 133-143.

[18] Shen, X., Tan, B., and Zhai, C. 2005. Context-sensitive information retrieval using implicit feedback. Proc. of SIGIR 2005, 43-50.

[19] Zigoris, P. and Zhang, Y. 2006. Bayesian adaptive user profiling with explicit & implicit feedback. Proc. of CIKM 2006, 397-404.

Search: From Information to Knowledge

Yan Qu
College of Information Studies
University of Maryland, College Park
yanqu@umd.edu

ABSTRACT

As more and more people use the Web as a knowledge base or a learning environment, it is important to provide easy access to existing knowledge structures on the web. This article advocates a new type of information seeking system that supports both topical search and the search for knowledge structure. Challenges and opportunities in designing such systems are discussed.

Keywords

Knowledge search, knowledge representation, the Web

1. INTRODUCTION

The Web is not only a huge information repository, but also a knowledge base or a learning environment where people learn new knowledge and find solutions to their real-life problems. Imagine students learn about scientific phenomena for their school projects; Patients and their family members try to understand difference between various treatments; intelligent agents track suspicious events and people to identify possible terrorist threats. As many web applications (particularly Web2.0 applications) have been developed to accumulate and synthesize knowledge (e.g. Wikipedia, Yahoo! Answers), we ask the question: How can we provide easy access to online knowledge, particularly knowledge structures? Can we make online knowledge structure searchable?

A search serving a learning or sensemaking purpose is fundamentally different from a search looking up for a piece of information. Most current search engines are designed for the latter. A keyword-based search mechanism (i.e. keyword-based input and keyword-based matching) with an appropriate ranking mechanism (e.g. PageRank) can provide users with information matching the query or information related to the corresponding topic. However, the need of a user doing a learning or sensemaking task is to gain knowledge. The user needs to grow a knowledge structure that incorporates the newly found information, such as concept maps explaining the relationships between concepts, probability networks enabling decision making, etc. The search system should provide not only topic related information, but also ideas for the creation and development of knowledge structure. Our previous study[12] showed that although users use the keyword-based search engines strategically to help themselves develop knowledge structures in learning and sensemaking tasks as they have no better choices, the design of the current information seeking systems is not satisfactory.

This article advocates a new type of information seeking system that supports both topical search and the search for structural knowledge representations. The author first examines the role of search in the process of growing a knowledge representation. This is followed by an analysis of the inadequacy of keyword-based search systems. Then, the author discusses the opportunities in designing searching systems that provide access to online knowledge structures.

2. THE ROLE OF INFORMATION SEEKING IN KNOWLEDGE REPRESENTATION CONSTRUCTION

Previous works on information seeking, learning and sensemaking have revealed a tightly coupled relationship between information seeking and knowledge gaining.

Dervin[5] proposed a general sensemaking model which is also regarded as an information seeking model, where information needs arise from the "Gap" between user's current knowledge and the knowledge needed to accomplish a task. People bridge the gap when they gather information to construct sense and move through the time-space context. According to this model, gaining knowledge and using knowledge to solve problems are the ultimate goals of information seeking. Information seeking is one step in the iterative cycle of knowledge gaining.

Knowledge is not only the passive understanding or interpretation of the world, but also the capability to act appropriately in the world. Instead of being some plain facts, knowledge involves structure or mechanism that enables calculation, reasoning, judgment, evaluation, decision making, etc. In their 1995 book, Nonaka and Takeuchi claimed "… knowledge, unlike information, is about action. It is always knowledge 'to some end.'" [9] Therefore, gaining knowledge is about structuring, changing, refining knowledge representation.

In Piaget's genetic epistemology theory [10], leaning consists of two types of process: assimilation - take information from the environment and encode it into the existing knowledge structure, and accommodation – change the knowledge structure to accommodate the external reality.

Similar processes were illustrated in Russell et al's Sensemaking model [14] as the Data Coverage Loop and the Representational Shift Loop (Figure 1). They defined sensemaking as "a process of searching for a representation (knowledge structure) and encoding data in that representation to answer task-specific questions". A sensemaker starts with an initial knowledge representation which

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Conference'04, Month 1–2, 2004, City, State, Country.

Copyright 2004 ACM 1-58113-000-0/00/0004 $5.00

he thinks could capture salient features of the information in a way that support the accomplishment of the task (the Generation Loop). Then he identifies information of interest and encodes it in the representation (the Data Coverage Loop). However, when the sensemaker's understanding of the sensemaking task grows, he may find that the initial representation is not adequate to characterize the sensemaking problem, which may impair the accomplishment of the sensemaking task. When this mismatch between his knowledge representation and the task (called "residue") becomes sufficiently problematic or costly (in terms of effort), the person is increasingly motivated to find a better representation, intending to reduce the cost of task operations (the Representational Shift Loop). The new knowledge representation is then used for encoding information, until sufficient residue builds up and yet a better representation is needed or the task can finally be satisfactorily accomplished.

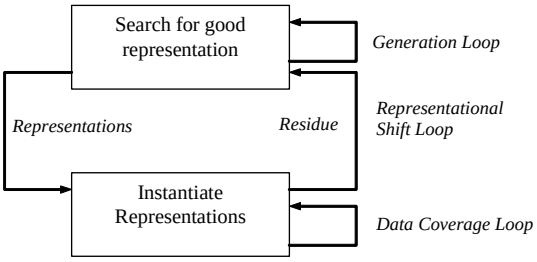


Figure 1. Representation development in Russell, et al's sensemaking model

Qu and Furnas [12] further examined where people get structuring ideas for knowledge representation, and the complex relationship between information seeking and representation construction. Other than generating structural representation using their existing knowledge relevant to the task, users also get ideas for structuring the representation from the outside information world (e.g. the Web). When they seek information, they also watch for new structure ideas or even ready-made chunks of knowledge structure in search results or Web pages navigable from the search results. They also use search as probes to validate those structure ideas. There is a bi-directional interaction between information seeking and representation construction. The existing knowledge structure (or structure ideas) shapes the information seeking by suggesting directions for future search and by helping people organize the search activities. Conversely, through various strategies, information seeking brings ideas and material for knowledge representation construction. If we consider learning or sensemaking as a process of seeking appropriate knowledge representation, then there is an information need to find new structure ideas, validate existing ideas, and find information to be added into existing structures. The search system, as an intermediary between the human and the external information world, should retrieve information that serves all these different needs.

Qu and Furnas' study suggested a variation of the traditional information seeking cycle, one that specifically highlights the seeking of knowledge structure (Figure 2). When people's existing knowledge representations are inadequate – incomplete or ill-formed generating "residue" in Russell et al's terminology, a special kind of "need for knowledge structure" arises. This is a need for changing, growing, or validating knowledge structures. To satisfy such need for knowledge structure, people often look for existing structures or ideas for structure instead of discrete information pieces in various information sources.

Although the popular keyword-based information seeking systems are not designed for the seeking of structure, people developed a strategy called Query-Initiated Navigation to deal with the problem. They issue some sort of query, usually just general or obliquely related keywords. Search results are returned which provide, not nuggets of information, but pointers into relevant websites, which are patches of interlinked and structured information on the Web. People then navigate those patches seeking ideas for changing, growing, or validating their current knowledge structures. They extract structuring ideas, either explicit fragments of structure for re-use, or perhaps just general structure-related inspirations. Here search is no longer a means to find information that directly satisfies users' information needs. Instead, the search leads people to an information patch where users can explore useful knowledge structures.

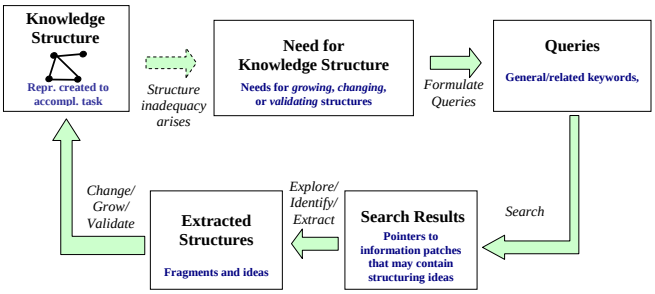


Figure 2. Knowledge Structure Seeking Cycle

3. THE INADEQUACY OF EXISTING INFORMATION SEEKING SYSTEMS

The knowledge structure seeking cycle reveals the inadequacy of existing information seeking systems in supporting the search for knowledge structures.

First a user's need for knowledge structure is hard to express using specific keywords. One reason is that the information needs for changing, growing, or validating structure are not matters of topic relevance, and are thus hard to capture using topic-expressing keywords; there is an inherent mismatch between the knowledge representation structure and the structureless bag of keywords that form a typical query. In many cases, diagrams or other graphical representations are more suitable for representing structures than words. Another reason keywords are problematic is that, when people seek for new knowledge, they may not be able to specify precisely what is needed. Belkin[1] called such a situation the "Anomalous State of Knowledge" (ASK), where people "recognized an anomaly in their state of knowledge of some topic, but they are unable to specify precisely what is necessary to resolve that anomaly". Moreover, keyword-based queries can hardly catch contextual information which is crucial for interpreting the need for knowledge structure, such as the user's task, the user's existing knowledge, the time and the place the information need arises, the socio-cultural or the socio-technical environment, etc.

Not only is this need for knowledge structure hard to express and interpret by existing systems, there is also a lack of effective

search mechanisms for finding and evaluating knowledge structures.

In a learning or sensemaking task, people look for semantic relationships or structures over different entities/concepts/topics, which are meaningful to human and can help them to build knowledge representations. However, knowledge structures exist at different granularities. For example, a sentence may contain a semantic structure over several entities, (e.g. A consists of B, C and D); a web page may contain semantic structures over many concepts and topics; the organization of web pages on different topics in a website may reveal structures over the topics; Images and diagrams sometimes also show knowledge structures. Current search engines are not able to identify, extract and integrate useful knowledge structures from existing information resources such as the Web.

In order to allow people to search for knowledge structures, in addition to identify meaningful structures, the information seeking systems also need to organize and index the structures to enable efficient access, to have algorithms to match knowledge structures to a user's information need, and to rank the structures based on their usefulness and quality. The current information seeking systems do not have such mechanisms to deal with structural information. This is partly due to the lack of awareness of people's structure seeking behavior, and partly due to the lack of advances technology for effective structure identification, indexing and ranking.

In addition to the technical difficulty, it is also hard for a system to judge which structure is useful to a user even if the information need can be captured perfectly because of the uncertainty in the information need. In the beginning of a search process, people usually do not know what they want except a general idea. Especially, they may have little idea about the structures they are seeking and know little about existing structures available on the Web. For example, a person starting to learn about digital cameras may not know any features of cameras yet. With such uncertainty in users' information needs, we need more intelligent systems to help the users to choose most useful knowledge structures for their tasks.

4. EXPLORATION OF DESIGNS OF KNOWLEDGE STRUCTURE SEEKING SYSTEMS

Although facing so many challenges, we have reasons to believe it is the right time to explore the design space of knowledge structure seeking systems because:

First, previous user studies showed that many people search for information to accomplish learning, sensemaking, and investigation tasks, in which they need to acquire knowledge structures. The inadequacy of the keyword-based search mechanism in supporting the acquisition of knowledge structure calls for changes in the design of information seeking systems.

Second, although the state of art technologies may still be inadequate to identify, index, and rank knowledge structures in information resources automatically, the existing technology might be able to facilitate users in their seeking for knowledge structures in an interactive manner. For instance, a search engine that allows input of certain context information (e.g. nature of the task) may tailer the search process to include more diverse information for exploratory tasks.

In this section, we will layout part of the design space of the knowledge structure seeking systems. We will emphasize those low hanging fruits that may lead to applicable research agenda.

4.1 Capture Different Types of Needs for Structure

First, we realize that there are different types of needs for structure, which should be handled differently by the system.

There are people who have little knowledge on a topic except a general topic name such as "camera". The system need to provide them a learning environment containing a proper knowledge structure of the topic. Notice that, showing search results without detailed descriptions of the concepts and structures is insufficient because without extra information to explain the knowledge representation, a user with little knowledge on the topic cannot judge the relevance and quality of the search results.

After people gain some basic knowledge about the topic, they may have preliminary structural representations on the topic. The structural representation may grow in different ways at this stage: to add more related concepts or sub-topics in the representation, to understand relationships between the concepts, to learn more about a specific sub-topic, etc.. For such a user, a similar but more complete knowledge representation should be helpful for the growth of his own knowledge representation. Detailed descriptions of the representation may not be necessary at this stage.

It's not easily to distinguish the different information needs automatically. The feasible way to handle this problem is to provide interactive conversations between the system and the user, let the user telling the system what they need. For example, the user can tell the system if his search is of the "lookup" nature or of the "exploration" nature, then the system can tailor the search algorithm accordingly. Tools can be designed to help users express their exiting knowledge structures in various representation forms such as concept map, tree, table, etc. Users' existing knowledge structures may also be detected from documents created or collected by the users, such as articles they have read and considered relevant. Users can also tell the system what type of changes they want on their existing knowledge structures (e.g. grow, refine, re-organize, etc).

4.2 Identify Knowledge Structures

One approach to facilitate knowledge structure seeking is to adopt structure search algorithms developed in hypertext researches [3][2]: a user specifies a desired topological structure and desired features of nodes and links in that structure. A system then looks for structures that match the required structure or match part of the required structure. Kaindl et al [7] had applied structure search to the web environment in which both hyperlink structures and page content are used in the search.

However, the structure search algorithms assume people are able to express the desired structure, and the algorithms search for the exact match of the desired structure. It may not suitable for knowledge structure exploration because 1) At the early stage of learning or sensemaking, a person may have little knowledge

structure to search upon; and 2) Different people may have different ways to structure knowledge representations on the same topic. Additionally, such structure search may suffer from vocabulary problems. The recall rate of structure searches may be even lower than that of regular searches because appropriate words are needed for multiple concepts in structure searches.

Other than direct structure search, systems can mine a data set to reveal knowledge structures in it. Researchers in the Natural Language Processing and Text Mining fields have long been interested in mining relationships within textual data. Linguistic models and machine learning techniques are used to automatically detect relations, patterns, and structures in textual data at various granularities. At the word level, relationships among lexical items can be detected using grammatical knowledge and statistical methods on large text corpora [6]. Moving up to the sentence/discourse level, relationships between sentences and discourse units can be detected using theories of rhetorical and discourse structures [8][11]. The discourse analysis can also be extended to cross-document relationship modeling and exploration [13]. There have been several works on detecting useful page-level structures on the web. For example, clustering technology are often used to reveal topics or subtopics in search results [4][15]. Worth mention is also the effort on the Semantic Web, which aims to create a universal medium for data, information, and knowledge exchange.

Unfortunately, at the current stage, we do not know much about how to index various knowledge representations (particularly when they are of different forms) and judge the relevance of a knowledge structures to a user's need for structural representation.

4.3 Support Query Initiated Navigation

As we mentioned in section 2, users often adopt a strategy called Query Initiated Navigation (QIN) to explore useful knowledge representations on the Web. An information patch (one or more websites) is suitable for QIN if it 1) has a network structure with good reversibility, 2) provides meaningful navigation cues, 3) suggests appropriate reading order, and 4) shows big picture or overview of the information patch. Algorithms or mechanisms that can identify information patches suitable for QIN will also be helpful in the search of knowledge structure. We can also add augment information structure on an information patch to help QIN. For instance, the system could enhance the navigation structure by highlighting paths that lead users to useful information in a website.

5. Conclusion

This article raises the question "how can we support the search for knowledge structure on the Web?", whose answer will help people in their learning and sensemaking tasks, and enhace the knowledge dissemination process in our society.

Other than exploit existing technologies to facilitate knowledge structure search as we discussed in section 4, two research directions are particularly challenging and exciting: First, in order to create effective knowledge structure search algorithms, we need to deepen our understanding on indexing and relevance evaluation of knowledge structures. Second, with the development of Web2.0 applications, there might be new opportunities of social computing mechanisms in the knowledge structure search on the Web. Human's cognitive system is more capable of recognizing structures and patterns than machines in many cases. Therefore, the approach that encourages a large group of people to identify, organize, and rank knowledge structure, and then integrates and utilizes the results might be fruitful.

REFERENCES

[1] Belkin, N. J. 1980. Anomalous states of knowledge as a basis for information retrieval. *Canadian Journal of Information Science*, 5, 133-143.

[2] Christophides, V. and Rizk, A. 1994. Querying Structured Documents with Hypertext Links Using OODBMS. *Proc. of the European Conference on Hypertext (ECHT '94)*, September 1994, pp. 186-197.

[3] Consens, M. P. and Mendelzon, A. O. 1989. Expressing Structural Hypertext Queries in GraphLog. *Proc. of the Second ACM Conference on Hypertext (Hypertext '89)*, Pittsburgh, PA, November, pp. 269-292.

[4] Cutting,D., Karger,D., Pedersen, J., and Tukey, J.W. 1992. Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections, *Proc. of the 15th Annual International ACM/SIGIR Conference*, Copenhagen, 1992.

[5] Dervin, B. 1992. From the mind's eye of the user: The sense-making qualitative-quantitative methodology. In J.D. Glazier, R.R. Powell (eds.), *Qualitative research in information management*, 6-84.

[6] Hearst, M. A. (1998) Automated discovery of WordNet relations. In: Fellbaum, Christiane, ed., *WordNet: An Electronic Lexical Database*, MIT Press.

[7] Kaindl, H., Kramer, S., Afonso, L.M. 1998. Combining Structure Search and Content Search for the World-Wide Web. *Proceedings of Hypertext 1998*: 217-224

[8] Marcu, D., & Echihabi, A. (2002) An unsupervised approach to recognizing discourse relations. *Proc. of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)* (pp 368-375).

[9] Nonaka, I. and Takeuchi, H. 1995. *The knowledge creating company: how Japanese companies create the dynamics of innovation*. New York: Oxford University Press.

[10] Piaget, J. 1978. *The Development of Thought: Equilibration of Cognitive Structures*. New York: Viking Penguin.

[11] Polanyi, L. (1988) A formal model of the structure of discourse. *Journal of Pragmatics*, 12: 601-638.

[12] Qu, Y., and Furnas, G.W. 2005. Source of Structure in Sensemaking. *Extended Abstract of Conference on Human Factors in Computing Systems (CHI'05)*, 1989-1992.

[13] Radev, D. (2000) A common theory of information fusion from multiple text sources, step one: Cross-document structure. *Proceedings of the 1st ACL SIGDIAL Workshop on Discourse and Dialogue*, Hong Kong, October 2000.

[14] Russell, D. M., Stefik, M. J., Pirolli, P., and Card, S. K. 1993. The Cost Structure of Sensemaking. *Proceedings of ACM INTERCHI'93*, 269-276.

[15] Clusty.com

Geography and Networks

Robert Reich
Me.dium
1050 Walnut
Boulder Co. 80302
646-369-9929
Robert@me.dium.com

ABSTRACT

Searching for relevant content on the public Internet has become an arduous task for many reasons, including but not limited to spam, poor content quality and information overload. Thus, a user searching “LCD monitor” might be overloaded with dozens of results of stores and price-comparison sites - significant time is then required by the user to sift through the content and locate what is relevant to their task.

In most approach’s today, the user must expend significant effort to seek out and identify relevant content. It would be desirable to build a system that could facilitate locating relevant content in a more natural an intuitive fashion. Me.dium has built such a system and this paper address a few of the key concepts and learning’s.

1. INTRODUCTION

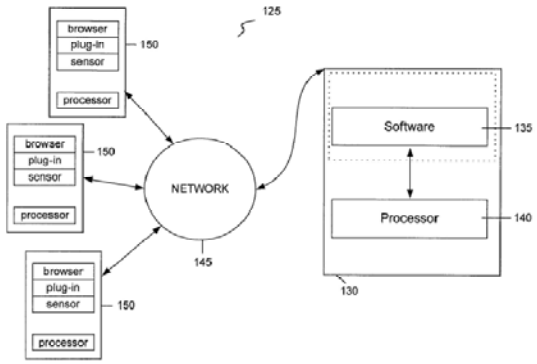
The concept presented in this paper aims to increase efficient utilization of information available in a content-based network, by dynamically forming unstructured ad-hoc communities.

The various functions and features of the system are aligned and configured based on a goal of transforming user interaction with a content-based network (for example, Internet browsing) into a communal, social experience, and then leveraging the dynamically formed contextual communities to facilitate sharing of highly relevant and vetted knowledge between users. The sharing can occur directly or indirectly.

The system also collects and analyzes user activities and reveals relationships between users, and between content that may not be apparent. The system further increases the efficiency and productivity of human-computer interaction by fostering dynamic sharing of context-relevant knowledge.

2. DATA COLLECTION (THE SENSOR)

Sensing is a key part of the system providing a simple way to gather relevant data in real time with minimal impact on the user. Sensing can happen either at the client side or the server side.



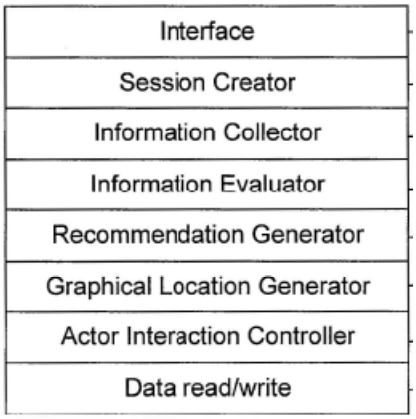
3. PRIVACY

The system provides an ability to experience people with similar interests online while protecting a sense of self and safety, both personally and collectively. This includes the ability to manage when personal identity is exposed. The user is given control over data collection (for example turning the sensor off if they engage in activities, which they don't want recorded). Optionally, the sensor may be controlled by white lists or black lists that are managed locally or remotely.

4. PERFORMANCES

Performance data in the system generally refers to an action performed by a user and the time at which the action was performed. Performance data is gathered for several reasons: to build community action information, to generate recommendations for a particular actor and to provide a logical way to organize the information so that it can be expressed visually.

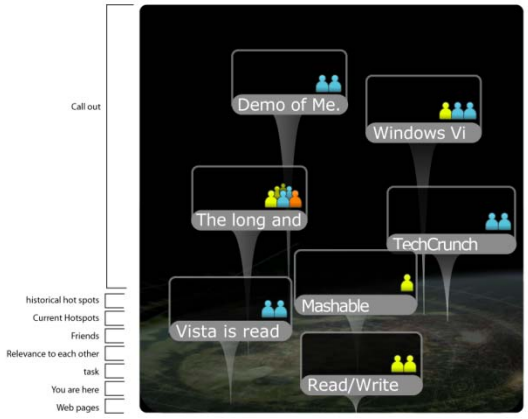
Performance data is collected from users (one to millions). This action data reflects the different actions that the various users typically perform. For example, the software could sense website navigation, navigation within a particular web page, pauses in activity, opening new windows, sending email, creating graphics, bookmarking, printing or any other action specific to a particular software program. Once the computer senses these actions and collects the data about these actions, that data is sent to a server. The server binds those actions to a particular time, adds appropriate meta-data and thereby creates a performance.



Certain action types may be more important than other action types depending on their context. Scrolling down a web page may be more important than copying content or playing a video may be more important than listening to an MP3 file.

For example, if the majority of the users watch less than 10 seconds of a 35 second video on www.cnn.com. The system can determine if it should recommend the video to the next user by comparing the new user's current and historical performances to that of the community.

Correlations are written out to a data file, which is represented as a graph. A graphical location generator uses the graph to generate backgrounds that convey the landscape of the internet. The background is sent to all users of the system. The generator also customizes the background by highlighting specific points of interest for each user in real time; these additional signals include but are not limited to: people, content and process.



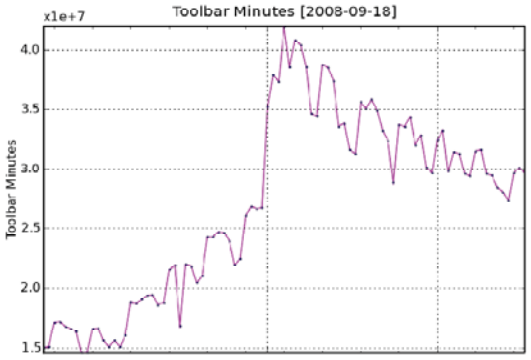
For example, the server could link a user's previous performance with an actor's most recent performance. This information could be stored in a table or other data structure. In essence, the server collects all the actor's performances over a period of time and uses those to generate recommendations for future actions based on community activity. Similarly, the list of performances performed by a particular actor can be added to the community information so that others can see the series of actions that this particular actor performed.

These actions can be displayed in different ways one is real time and another may be in a list.

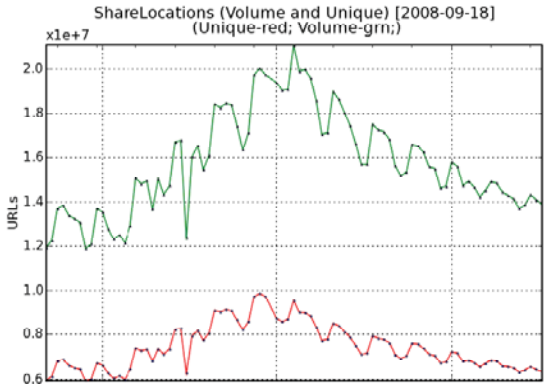
5. CONCLUSION

The system has been operating for a little over 12 months and data collection and ranking is very strong for certain types of queries.

For example toolbar minutes collected can be as high as 40 million minutes per day



URL volume per day has reached 20 million, which could contain as many as 600,000 unique domains.



Several other areas being researched include the tall head (most popular web pages) and the long tail (unique one off web pages), plus more advanced ways to incorporate the real time nature of the system into the graphical representations.

6. ACKNOWLEDGEMENTS

Founding partners at Me.dium

References

United States Patent 20070192461 - -Robert Reich and Peter Newcomb. SYSTEM AND METHOD FOR DYNAMICALLY GENERATING AND MANAGING AN ONLINE CONTEXT-DRIVEN INTERACTIVE SOCIAL NETWORK

What might users be learning from the system?

Catherine L. Smith
Rutgers University, SCILS
4 Huntington Street
New Brunswick, NJ USA
csmith@scils.rutgers.edu

ABSTRACT

Perhaps the most reliable characteristic of any common web search system is the correlation between an item's position on a results list and the probability that the item will be useful. Several recent studies suggest that search system users have learned this property of ranked lists, and have developed routine procedures (habits) for interaction during search. This paper briefly reviews those studies. The paper then presents additional experimental evidence illustrating that while searchers rely on position as an indicator of an item's value, they also alter their behavior when that evidence becomes unreliable. The paper concludes by arguing that effective mechanisms for assisting searchers will invoke and guide the development of new procedures (habits) for non-routine, complex search. Such a system would, in effect, help its user learn how to search.

1. INTRODUCTION

An information need is generally part of some larger goal, which may be composed of a complex set of sub-goals [4]. Once a searcher has selected an interactive search system for solving the problem of an information need, the user must solve the sub-problem of inducing the system to display the desired information. The sub-problem is the subject of this paper.

One design approach for supporting information search is a system that learns the relationships between queries and the documents that meet the information needs expressed by queries. In this approach, the *system* learns the relationships. This approach is effective for recurring, simple needs because the system has many examples of query/document pairs. In more complex implementations of this idea, the system learns by using additional evidence of common contexts or user states. For complex, non-routine needs, where a query/document pair is rare or unique, the system has little information with which to learn. In that case, this design solution may fall short of delivering needed performance, and search becomes difficult.

An alternative design approach focuses on supporting the *user's* learning of the relationships between queries and search results, so that the user can produce a more effective query. Many of these ideas are reviewed in Jansen [7] and Jansen & McNeese [8], where the authors also discuss an evaluation of their AI2RS search assistant system. In another example, Anick & Kantamneni [1] discuss Yahoo's recent implementation of a query term suggestion assistant. Evaluations of these systems reveal that searchers often overlook or ignore the assistance provided. Searchers chose to solve their search problems using their own routines.

Evidence from recent studies suggests that searchers have developed their own routine procedures for interaction. Searchers' "habits" rely heavily on the regularities of ranked lists. Users have learned the correlation between the probability that an item will be useful and the position of the item on the list. Other research has found that users increase the pace of query entry when faced with a poorly performing system. These findings suggest that searchers continue to use their habits of quickly scanning a list when faced with a difficult search. These findings are review below.

2. RECENT STUDIES

Effect of item ranking. Eye-tracking studies have revealed important information about how users interact with ranked results lists. Recent work has described the order in which users examine items on a list and the amount of visual attention (measured as fixation duration) given to items. In other work these measures have been related to click-through probabilities. Much of this work has been focused on understanding how well click-through indicates document relevance, but the results also suggest that users have developed strong habits for interaction with results lists.

It is well established that searchers use item ranking as a cue to the relevance of an underlying information source [4, 5, 6, 9, 10, 12]. The top two items in a retrieved list are fixated more frequently and are fixated for a longer period than any other item positions. The top item is particularly privileged: it is clicked with the highest frequency and it is more likely that it will be clicked, even if the 2nd item is relevant and the 1st one is not ([7], but see [5] on navigational search). The tendency to click on the 1st item has been termed a click-through "trust bias".

Findings related to the *order* in which users scan items lower on the list are not as well established. While most studies suggest that users scan retrieved lists from top to bottom, it is also clear that visual attention is not completely locked into this process. Joachims, et al. [9] found that for half of cases, the item directly below a clicked item had been scanned prior to a click. In a detailed analysis, Lorigo, et al. [12] found that not all subjects used a linear (top down) scanning strategy exclusively. When their subjects clicked on an item in the list, two thirds of the time all of the items above the clicked item had been scanned at least once. However, only one fifth of scan-paths analyzed were *strictly linear*, where items were scanned in the exact descending rank order (with no skips or scans of a previously scanned item). Klocker, Wirschum, & Jameson [10] also found that many subjects (35% in one experiment, and 48% in another) employed what they termed a *breadth-first* scanning strategy in which a subject's gaze returned to click on a previously scanned, higher ranked item.

The above findings indicate that people have learned the dominant statistical property of ranked lists: over the long run,

the probability that an item will be useful is proportional to its position on the list. Searchers have developed visual scanning patterns that reflect this. Their visual attention and interactions with the list are focused at the top.

Effect of system performance. Several recent studies have reported effects of system performance on search behavior. Joachims, et al. [9] examined the effect of item ranking on visual fixation and click-through behavior; the study is discussed further in Lorigo, et al. [11]. Three systems were used in the study: a standard Google system (*normal*) and two degraded systems. The degraded systems were produced by manipulating item rankings using one of two interventions. Results for the two methods are discussed in turn.

In the first type of performance intervention, the order of the first two items was *swapped*, so that the item estimated by the system to be the most likely match appeared as the 2nd item on the list. The authors also investigated the effect of the swap by comparing behavior in the normal and swapped conditions. When the 1st item was more relevant than the 2nd and the searcher clicked, in both conditions subjects were very likely to click the most relevant item (95% of clicks in normal condition and 94% of clicks in the swapped condition). When the 2nd item was more relevant than the 1st and the searcher clicked, in both conditions subjects were *less* likely to click the relevant item (44% of clicks in normal condition and 47% of clicks in the swapped condition). It appears that the searchers did not detect the swapped condition, and that they proceeded to search without changing their behavior.

In the second intervention, the order of the items on the list was *reversed*. Each list contained 10 items. After reordering, the item estimated by the system to be the best match to the query appeared as the 10th item on the list. Searches conducted in the reversed condition were compared with those completed in the normal condition. Subjects in the reversed condition changed their behavior. They scanned significantly more items (3.8 abstracts vs. 2.5), took more time to scan the list (11 vs. 6 seconds), were less likely to click any item on the list (.64 clicks vs. .80), and were more likely to click on an item at a lower rank (average rank of click 4.03 vs. 2.66). Subjects using the reversed system did not, however, overcome their rank-based bias. They were more likely to click one of the first 5 items in the list, and less likely to click one of the last 5 items. These subjects were also less likely to complete their task as successfully as those who used the standard system (62% vs. 85%). The above results indicate that searcher's detected the reversed condition and adapted their behavior. While they did not reach the level of success possible with the normal system, they did succeed on a small majority of searches.

For a complex search task with no time limit, Smith & Kantor [13] compared searches conducted using a system with standard performance to those conducted using two systems with intentionally degraded performance. The systems were degraded by displaying results from very low positions on Google's results lists, however the order of the lists was not altered. They found no significant difference in the success of searches conducted in each condition. They did find a significant increase in rate of query entry for searches conducted using the consistently bad system. The findings suggest that when a system performs poorly, people are able to quickly detect the low value of the results page and quickly enter a new query.

Together these findings indicate that searchers have the ability to alter their behaviors when confronted with an aberrant list. However, it is clear that they often fail to do so. The searcher makes a decision between investing time on the existing list, and investing time on the production of a new query. The question of which action is optimal under which conditions is an empirical question, not addressed here.

3. ADDITIONAL EVIDENCE

Data collected in the experiment reported in [13] further illustrates the effect of system performance on rank-based bias. The data reported are from the 2nd block of the experiment (the reader is referred to that paper for additional details). 36 subjects were recruited on the campus of a large east-coast university. 12 subjects were assigned to each of 3 groups. Each group searched using a different version of Google. The systems were the same in all respects except for the retrieval performance of each version. A *control* group used a version that displayed standard results lists. Subjects in the other two groups received results lists that were intentionally degraded. The *CLR* group received results that always started with the 300th item on the Google list (consistently degraded). The *ILR* group received results that started at various ranks between the 1st and 300th (inconsistently degraded). Each results page displayed a maximum of 20 items, with no option to continue to the next page of results. Each subject completed 4 topic searches, for a total of 48 searches by each group. Each topic was searched the same number of times by each group. Subjects were told that they needed to find as many good information sources as possible for a hypothetical "boss", and as few bad sources as possible. The topics were complex and informational. A small check box was displayed next to each item in the Google list; the checkboxes were used to indicate each good information source found. The searches reported here occurred after subjects had completed a prior block of 4 searches, using the standard system. Advertisements were removed from all results. The system recorded the position of each item on each list, and each item that was identified as a *good information source* by subjects. Every item identified as 'good' by a subject was subsequently judged as to its 'goodness' by the researcher, who was blind to the conditions under which the item was identified.

The data are displayed in 4 graphs (below). Item ranks 1 – 20 are along the abscissa of each graph, where item 1 is the item displayed at the top of a 20 item list. The graphs depict system performance and subject behavior. The data reported has been aggregated for each group. For each rank, the data point includes all item displays, for all results lists returned during the 48 topics searches conducted by the group.

Figure 1 graphs, for each group, the probability that when an item was displayed, it was a good item (as judged by the researcher, see above). The effects from the manipulations of the starting ranks are clear. For 19 of 20 ranks, subjects who used the standard system were more likely to receive a good item than were those in the CLR group. For 15 of 20 ranks, subjects who used the standard system were more likely to receive a good item than were those in the ILR group.

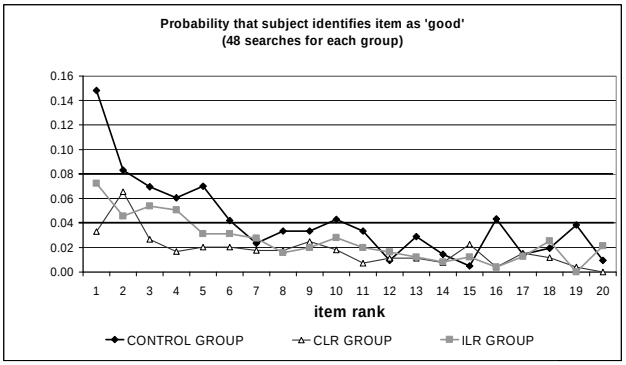


Figure 1. Performance of experimental systems

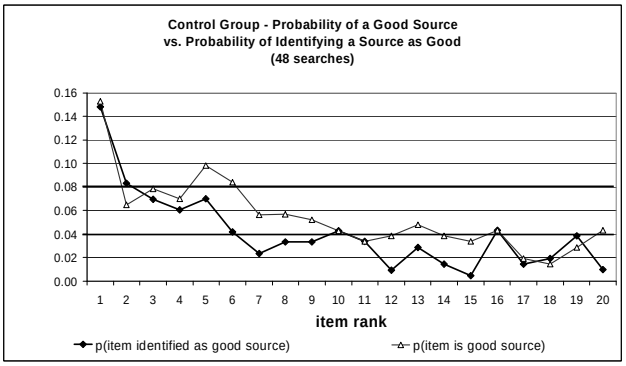


Figure 2. Control Group – System Performance and Subject Behavior

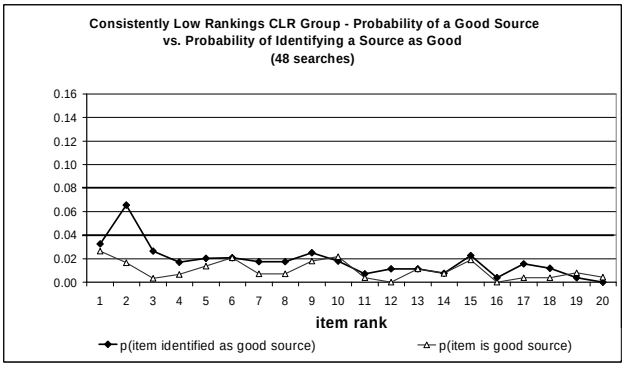


Figure 3. CLR Group – System Performance and Subject Behavior

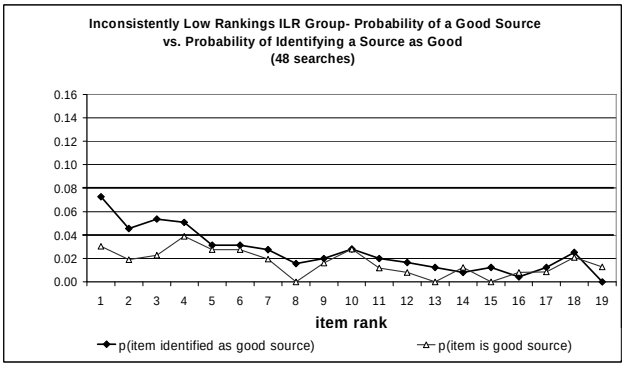


Figure 4. ILR Group – System Performance and Subject Behavior

Figures 2, 3, and 4 display results for the control, CLR and ILR groups, in turn. Two probabilities are graphed for each group: 1) the probability that an item displayed is a good item (as above) and 2) the probability that a displayed item was identified by subjects as a good item (but not necessarily judged to be good). For subjects using the standard system, the probability of identifying the top-ranked item as 'good' is

essentially equivalent to the probability that the item was good. However, at almost all ranks below the first rank, the probability of identifying an item as 'good' is *lower* than the probability of a good item. Subjects in the control group failed to identify good items lower on the list. Subjects using the degraded systems were sensitive to the poor performance they encountered. They were less likely to identify an item as 'good' than were those who used the standard system. Subjects in the IRL group did not, however, fully abandon their rank-based habits. As was the case for searches conducted using the standard system, the probability of identifying an item as 'good' was lower for items lower on the list. In addition, the probability that an item was *identified* as good was higher than the probability that the item *actually* was good. Subjects either lowered their standards for items at the top of the list, or were over-reliant on the evidence supplied by the ranking. Subjects in the CLR group appear to have recognized the low quality of the first item on the lists they received. It appears that the group may have shifted their trust bias to the second item on the list. Other than this anomaly, subjects in the CLR group appear to have given up most of their rank-based bias. This illustration suggests that rank-based bias is affected by large and consistent changes in system performance.

4. DISCUSSION

The findings above imply that search systems have taught their users to rely on the structure of ranked lists during interaction. Searchers have learned the lesson well. Their habits may, however, be sub-optimal for complex, non-routine search. The evidence reviewed and reported above suggests that searchers must experience a sufficient level of difficulty before behavior is altered. As Jansen & McNeese [8] and Jansen [7] point out, if the system intervenes at the appropriate moment during difficulties, searchers are more likely to be receptive to assistance. Beyond this, however, the ideal system would not simply assist, but would guide and support the development of alternative "habits" for complex search. Of course, a habit is a routine that is reliable in a broad range of cases. Further research is needed in order to understand what the optimal habits are for complex search, and how different those habits are from the adaptive behaviors searchers have learned to use.

Another aspect of the problem is that the utility of new habits must be readily available to the searcher. A spelling suggestion mechanism is a simple example of an assistive device that appears to meet this criterion. The value of a correctly spelled term is readily available to the user in the quality of the results list, and the cost of using the suggestion is low. In this sense, the system teaches its user the utility of correct spelling. The ideal system offers spelling support when, and only when, there is a sufficient chance that the user will correctly predict its usefulness. For a user who has learned *how* to predict the usefulness of correct spelling, the ideal system would offer spelling support before an initial query is entered, so that the

routine of considering spelling is integrated into the user's solution to every search problem.

Spelling mechanisms are a very specific form of query-term suggestion. Spelling services solve a highly routine search problem, one which can be encountered in any type of complex or simple search. While other forms of query-term suggestion have been developed, they are not necessarily designed to solve *routine* search problems. Further research is needed to identify routine search problems (to be clear, this means *habits of interaction*, not routine of information needs). If a mechanism such as query-term suggestion is to be useful, the system must teach its user how to reliably predict the expected value of the results produced by the mechanism.

5. REFERENCES

[1] Anick, P., & Kantamneni, R. (2008). A longitudinal study of real-time search assistance adoption. *Proceedings of 31st SIGIR Conference*, Singapore. 701-702.

[2] Card, S., Moran, T., & Newell, A. (1983). *The psychology of human-computer interaction*. Hillsdale, NJ: Lawrence Earlbaum Associates.

[3] Clarke, C., Agichtein, E., Dumais, S., & White, R. (2007). The influence of caption features on clickthrough patterns in web search. Amsterdam, The Netherlands. 135-142.

[4] Cutrell, E., & Guan, Z. (2007). What are you looking for? An eye-tracking study of information usage in web search. *Proceedings of SIGCHI Conference*, San Jose, California. 407-416.

[5] Guan, Z., & Cutrell, E. (2007). An eye tracking study of the effect of target rank on web search. *Proceedings of SIGCHI Conference*, San Jose, California. 417-420.

[6] Granka, L., Joachims, T., & Gay, G. (2004). Eye-tracking analysis of user behavior in WWW search. *Proceedings of 27th SIGIR Conference*, Sheffield, UK. 478-479.

[7] Jansen, B. (2005). Seeking and implementing automated assistance during the search process. *Information Processing & Management, 41*, 909-928.

[8] Jansen, B., & McNeese, M. (2005). Evaluating the effectiveness of and patterns of interactions with automated searching assistance. *Journal of the American Society for Information Science and Technology, 56*(147), 1480-1503.

[9] Joachims, T., Granka, L., Pan, B., Hembrooke, H., & Gay, G. (2005). Accurately interpreting clickthrough data as implicit feedback. *Proceedings of 28th SIGIR Conference*, Salvador, Brazil. 154-161.

[10] Klockner, K., Wirschum, N., & Jameson, A. (2004). Depth- and breadth-first processing of search results lists. *Proceedings of 27th SIGCHI Conference*, Vienna, Austria. 1539.

[11] Lorigo, L., Haridassan, H., Brynjarsdottir, H., Xia, L., Joachims, T., Gay, G., et al. (2008). Eye tracking and online search: Lessons learned and challenges ahead. *Journal of the American Society for Information Science and Technology, 59*(7), 1041-1052.

[12] 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. *Information Processing & Management, 42*, 1123-1131.

[13] Smith, C.L. and Kantor, P.B. (2008). User Adaptation: Good Results from Poor Systems. *Proceedings of 31st SIGIR Conference*, Singapore. 147-154.

Challenges for Supporting Faceted Search in Large, Heterogeneous Corpora like the Web

Jaime Teevan
Microsoft Research
Redmond, WA, USA
teevan@microsoft.com

Susan T. Dumais
Microsoft Research
Redmond, WA, USA
sdumais@microsoft.com

Zachary Gutt
Microsoft Corporation
Redmond, WA, USA
zachg@microsoft.com

ABSTRACT
Faceted search systems help people find what they are looking by allowing them to specify not just keywords related to their information need, but also metadata. While such systems hold great potential and have been successfully used in vertical domains, there are many challenges in extending them to large, heterogeneous collections like the Web, corporate intranets, or federated search engines that access many different data silos. In this position paper we discuss the challenges in greater detail. Those that we have identified stem from the fact that such datasets are 1) ***very large***, making it difficult to assign quality meta-data to every document and to retrieve the full set of results and associated metadata at query time, and 2) ***heterogeneous***, making it difficult to apply the same metadata to every result or every query.

Categories and Subject Descriptors
H.5.4 [Information Systems]: Information Interfaces and Presentation (e.g., HCI) – Hypertext/Hypermedia: User issues.

General Terms: Human Factors, Measurement.

Keywords: Faceted search, filtering, metadata, Web search.

1. INTRODUCTION
The term *facet* means "little face" and is often used to describe one side of a many-sided object, especially a cut gemstone. In the information science literature, the term has been used to refer both to the organization of information (faceted classification), and to interfaces that provide flexible access to that information (faceted search). An important motivation for faceted systems is that any single organizational structure is too limiting. Multiple independent facets provide alternative ways of getting to the same information, thus supporting a wider range of end-user tasks and knowledge. Interfaces to faceted information usually include capabilities for structured browsing (or faceted navigation), and some offer search capabilities as well. In this paper we explore some of the challenges involved in developing faceted search systems for large, unstructured and heterogeneous collections.

The principles of faceted organization are widely applicable. Each facet represents a dimension that can be used to organize the information (e.g., topical category, price, manufacturer, color, etc.). Each facet has a name or label, which can be alphabetic, numeric, categorical, continuous, etc. Facets can be organized hierarchically or as a flat list. Every item in the collection is assigned one or more values on each facet. A probability or confidence can be associated with each value, as often happens when values are assigned automatically, although interfaces that expose this are rare.

Faceted search systems augment full-text search capabilities by providing additional structure to support query refinement or results presentation. Often when people search for information, they prefer to specify as little as necessary in their query to find what they are looking for [1, 2, 8]. Rather than fully specifying their target up front, searchers often prefer to interact with the results to refine their query as necessary. For many search tasks, an initial query is sufficient. When modifications are necessary faceted search provides an easy way for people to further describe what they are looking for. For example, if a person were looking for a $200 red digital camera, instead of typing "$200 red digital camera" into a commerce site's search box, that person may first search for "cameras", and then refine the query by selecting the "digital camera" category, the appropriate price range, and the camera color of their choice. This type of faceted search interaction, which combines full-text search and metadata browsing, has been successfully used in many search verticals, and is commonly seen in e-commerce Web sites, desktop search applications, library databases, etc.

However, there are many challenges to extending the successes of faceted search to large, heterogeneous corpora like the Web, large corporate intranets, or federated search engines that access many different data silos. In this paper, we first summarize some of the lessons learned from previous successful implementations of faceted search in more limited domains, and then discuss some of the challenges faced when scaling up to large, heterogeneous applications.

2. RELATED WORK
Several examples of faceted search systems have been discussed in the research literature, including faceted metadata systems for images [1], movies [5], houses [6], and desktop content [1]. In addition, many Web sites use faceted search to provide access to their content. Examples include: library catalogs (e.g., www2.lib.ncsu.edu/catalog), images (e.g., gettyimages.com), and shopping sites such as BestBuy (bestbuy.com), Home Depot (homedepot.com) and eBay (ebay.com).

Previous research has examined a number of the challenges for developing effective faceted search systems. For example, one issue is how best to represent continuous dimensions. A popular approach is to group continuous facets like "Price" into bins (e.g., $1-$100, $101-$200) that can then be selected. However, bins do not allow users to capture finer distinctions. Shneiderman [6] developed richer interaction techniques that use sliders to highlight ranges of interest and dynamic query techniques to update the display of matching results in real-time.

Another challenge that has been explored is how facets should be combined. Different facets can potentially be specified in any

order and combined to identify a set of items using the full power of Boolean logic. Enabling users to richly express what they are looking for without overwhelming them is an important design goal. In practice, most systems use AND to combine selections from different facets (e.g., red AND $200), and OR to combine selections from the same facet (e.g., (red OR black) AND $200). Hearst [4] provides a nice summary of emerging best practices in user interface design for faceted search, including which facets to show (and how to provide access to others), graphic techniques to display facet labels and matches, and breadcrumb design to indicate the current query terms and facet selections.

In this paper, we discuss additional challenges that may be encountered when applying faceted search to large, heterogeneous corpora. We highlight three issues (generating metadata when it is not explicitly available, identifying which facets to use, and providing quick and accurate metadata profiles), and we look forward to discussing additional issues with workshop attendees.

While there have been attempts to structure the content of the Web using a topic hierarchy like Open Directory (dmoz.org) or the Yahoo! directory in its early days, such systems reflect only a single facet (topic), and the content has not always been tightly integrated with full-text search. Similarly, many search engines provide related searches that allow users to specialize or generalize their requests, but again this exposes only a single dimension (words, which are different in many ways to more traditional facet organizations). Here we focus on the issues related to the tight integration of full-text search and rich faceted navigation.

3. CHALLENGES

The challenges we have identified to applying faceted search to domains like the Web stem from the fact that such datasets are very large and heterogeneous. Because they are very large, it is difficult to assign quality meta-data to every document in the collection and to retrieve the full set of results and their associated metadata at query time. And because they are heterogeneous, it is difficult to apply the same facets to every result or every query. In this section we discuss these issues in greater detail.

3.1 Automatically Generated Metadata

Most domain specific search engines have relatively clean metadata associated with the items in their corpus. For example, commerce search engines tend to be built upon databases with accurate price and brand information. Because other corpora of interest, such as intranets or the Web, do not have pre-assigned metadata, many facets are likely to be assigned algorithmically. This means that some of the metadata may be wrong or have a probabilistic value assigned for it.

When determining how to tune an algorithm that automatically assigns metadata for use in faceted search, it is important to balance the cost of mistakenly assigning a metadata attribute to an information item with the cost of not assigning a piece of metadata to an item when it should be. If selecting a facet yields a lot of unexpected and irrelevant results, users may not find the selection to be worthwhile. On the other hand, if selecting a facet causes many relevant results to be removed from the result set, users may find the risk of missing something valuable to be too high to use the system. Our hypothesis, given the importance of precision in Web search, is that it is better to be accurate than comprehensive, but the right balance surely depends on many factors, including the user's information need, context, and the facet in question.

Rather than making a binary decision that a facet applies to an information item or not, a score can be assigned to indicate the confidence in the assignment. There may be ways to surface this confidence in the assignment of facet labels in a way that makes users comfortable. One possibility is to use a slider that starts with the items that have the highest confidence associated with them and gradually add less certain items. Another place where people appear to have some tolerance for ambiguity is in the ranking of Web search results. Users understand that relevant results are ranked first, less relevant results are ranked later, and that this ranking may or may not be perfectly accurate. Using metadata to support different rankings, rather than to merely filter results, may provide value in some cases. As an example, a person looking to buy a digital camera could search for "digital cameras" and then select "commercial sites" not to filter the results, but rather to rank the results so that those most likely to be commercial are listed first.

Ranking result sets by metadata may prove value, too, in enabling people who are searching very large datasets to better access the long tail. If filtering search results preserves the initial query-based ordering, valuable data that is relevant but ranked relatively low may never be seen. For example, a person who searches for "restaurants" and then filters by "near me" may not want to see the hundreds of restaurants near them ordered by how closely they match the query "restaurants", but rather prefer to see the results ordered by those closest to them.

Another challenge to automatic facet generation is that there are a very large number of different types of facets that one could automatically extract about documents, from simple indications of the presence or absence of a keyword in a document (e.g., "camera"), to much more complex (e.g., synthesizing all of the keywords in the document to determine that it is about "photography"). It is not obvious what level of granularity is appropriate to expose. People may want to interact with fine grain, simple facets that are particularly accurate (e.g., we know for sure if the word "camera" appears in a document), or with concepts that may be less accurate but more expressive. When working with a large number of facets it is also important to identify which facets to surface for a particular query or result set, as we discuss in the next section.

3.2 Identifying which Facets to Surface

Many domain specific search engines, such as ones designed to support commerce searches, recipe searches, or image searches, only need support a relatively narrow range of user tasks. In these cases, it is easy to predict which facets will be the most useful for the searcher. In the case of commerce site, price and brand may be particularly useful, while in recipe search, the ingredients or course may be most useful.

On the other hand, people use more general search engines for a much wider range of complex tasks. On the Web, people conduct research, plan trips, purchase items, and find new jobs using search engines. Similarly, on a corporate intranet people may search for experts, colleague contact information, corporate policies, or valuable research all with the same search engine. When the queries applied to a search system are varied in intent it is unlikely that all facets will apply equally well to all queries. While there may be some commonly useful facets that are always worth displaying, others may need to be selected for display on-the-fly. This raises a number of interesting questions, such as how many facets should be display in a given context, in what order, and, most importantly, how should the most relevant facets be identified.

Facet identification can happen manually or automatically. In the case of manual identification, easy ways must be developed for the user to browse through a large list of potentially irrelevant facets to find the ones they want. One way to winnow this list down may be to eliminate facets that contain no results for the current query. However, as we will discuss later, even this can be a challenge with very large collections of information.

In many cases it may be that people prefer to have the most relevant facets identified for them. The initial query and result set could suggest valuable facets. For example, facets that partition the result set well, facets that are commonly selected for a query, or facets that appear more often than expected may be particularly worth displaying. However the facets that are optimal from a statistical perspective may not correspond to those that the user can best recognize or specify. Additional information may be provided by the user implicitly as they reformulate their query and interact with the result set and the facets. Facets that a particular user has previously found useful may be particularly valuable for that user.

One challenge in dynamically identifying the most appropriate facets for each query and associated result set is that consistency and predictability will be reduced. A more consistent ordering of facets may be useful so that users always know where to find the facets they expect. Or, building on the dynamic menu example, it may be useful to copy split menus [7] and preview a few facets that are particularly likely to be useful while still providing more predictable access to the entire set. Another way to provide some consistency within a task type would be to group facets and trigger the entire group for appropriate queries. For example, a commerce query could trigger a set of facets with price and product information, while a recipe-related query could trigger a set of facets with course and ingredient information.

3.3 Hard to Accurately Preview Facets

Another challenge with supporting faceted search over very large or distributed corpora is that the search engine must be able to quickly compute (or estimate) the facet values for every result that matches a particular query. A search for "tom jones", for example, may returns tens of millions of documents. Most commercial search engines examine only a subset of the possible matches in detail, so it may be difficult to compute the full distribution of facet values for all matching items.

The difficulties in knowing detailed information about the complete result set makes facet identification harder, and potentially more dynamic since the result set available for facet identification changes as the user interacts with it. It can also make previewing facets to give users an idea of what to expect when they select a particular facet challenging. Many faceted search systems preview how many results will be returned if a particular facet is selected. For very large databases, it probably makes sense to abstract this preview to a few discrete buckets (e.g., *one*, *a few*, and *many*), but even a preview intended only to indicate the presence or absence of a result with that facet may be inaccurate. Understanding how to develop algorithms to more accurately predict the distribution of metadata values for a dynamic subset of items (namely those returned for the current search) is a valuable direction for future work.

4. CONCLUSION

Faceted search systems have been used successfully for many vertical applications, including e-commerce, image databases, and library catalogs. In this paper we have discussed some of the challenges that must be faced when considering how to apply ideas from faceted search to support access to large, heterogeneous collections, such as general intranet or Web content. These challenges include how to generate metadata when it is not explicitly available, how to identify which facets to display for a query (and associated result set), and how to provide quick and accurate metadata profiles of the content.

REFERENCES

[1] Cutrell, E., Robbins, D., Dumais, S., and Sarin, R. (2006). Fast, flexible filtering with Phlat. In *Proceedings of CHI '06*, 261-270.

[2] Downey, D., Dumais, S., Liebling, D., and Horvitz, E. (2008). Understanding the relationship between searchers' queries and information goals. To appear in *Proceedings of CIKM'08*.

[3] Dumais, S. (2008). Faceted search. *Encyclopedia of Database Systems.* M. T. Ozsu and L. Liu (Eds.) Springer 2009.

[4] Hearst, M. (2006). Design recommendations for hierarchical faceted search interfaces. In the *SIGIR 2006 Workshop on Faceted Search.*

[5] Koren, J., Zhang, Y, and Liu, X. (2008). Personalized interactive faceted search. In *Proceedings of WWW '08*, 477-486

[6] Shneiderman, B. (1994). Dynamic queries for visual information seeking. *IEEE Software, 11(6),* 70-77.

[7] Sears, A. and Shneiderman, B. (1994). Split menus: Effectively using selection frequency to organize menus. *TOCHI, 1(1),* 27-51.

[8] Teevan, J., Alvarado, C. J., Ackerman, M. A., and Karger, D. R. (2004). The perfect search engine is not enough: A study of orienteering behavior in directed search. In *Proceedings of CHI `04*, 415-422.

[1] Yee, P., Swearingen, K, Li, K., and Hearst, M. (2003). Faceted metadata for image search and browsing. In *Proceedings of CHI '03,* 401-408.

Novel User Interfaces via Model-Mediated Information Retrieval

Earl J. Wagner, Jiahui Liu and Larry Birnbaum
Intelligent Information Laboratory
Northwestern University
Evanston IL USA
{ewagner, j-liu, birnbaum}@cs.northwestern.edu

ABSTRACT

Using content-specific models to guide information retrieval can provide richer interfaces to end-users in both navigating news articles and learning the context of news events. We present Brussell, a system that uses semantic models of news event situations to perform anticipatory information retrieval, organize extraction results and present a novel interface for navigating among the milestone events of a situation.

1. INTRODUCTION

People browse the web not only to search for specific facts, but also in "'building a picture' of an organization, topic or person." [11] However, the nature and specific *kinds* of "big picture" views that might benefit information gatherers, and how software might be constructed to support their elaboration, has not received nearly as much attention as search more narrowly construed.

The need for a "big picture" view is particularly acute when reading news. An article may cover a new event involving organizations and individuals previously unknown to the reader. Or the reader may be familiar with the event participants, but not with the overall situation involving the event—where by *situation* we mean a limited sequence of causally-related events, such as all of the newsworthy actions in a lawsuit. For example, the dismissal of a lawsuit follows the filing of the lawsuit and both are part of a particular lawsuit situation.

In establishing the context of a new event, news articles reference previous events. Often these events are related to the topic of the current article by being part of the same overall situation - perhaps an earlier event in the situation, such as the filing of the suit. Or it may reference other similar or related situations. A similar lawsuit may be taking place in another locale. Related lawsuits include a suit acting as a case precedent, or other suits involving some of the same participants, such as other suits against the defendant.

All of these relationships are part of the situational context that the user draws upon in making sense of the events the article describes. This context gives rise to specific questions, such as:

- What happened in this situation?
- What happened in the other situations referenced in this article?
- What other similar and related situations have these participants been involved in?

Neither conventional news web pages nor current browser software provides content-specific support for answering these questions, however.

Some online news sources offer links to related pages, but these are frequently irrelevant or out of date. An article web page about the filing of a lawsuit isn't typically updated to link to coverage of the lawsuit's dismissal. Some articles link previous-event textual references to earlier articles, though these links must be added manually.

Without an in-page link, to answer her natural questions, the user must find related articles manually. She must identify relevant terms such as entity names and situation keywords. Then she must cut-and-paste them into a news search engine. Finally she must sort through lists of results to find relevant articles. These steps make for an inconvenient process familiar to anyone who reads news on the web. Even news timelines provided by advanced search engines are unable to provide content-specific overviews of a situation in accordance with the user's expectations of how it begins and continues.

Existing automated approaches typically offer support through domain-independent methods, such as by clustering articles based on term frequencies, or summarizing multiple articles about the event. These approaches don't leverage a user's expectations, however, for how the situation has unfolded *causally* and how it will proceed. For example, a lawsuit that begins with a high-profile filing may end with a low-profile settlement. Although a user expects the lawsuit to end in one of several ways, domain-independent systems do not and may miss these more obscure events. A domain-specific approach is necessary to support users' expectations for how events relate in a situation and thus enable new kinds of user interaction.

We present ***Brussell***, a system that performs anticipatory information retrieval and model-based information extraction to support the user in exploring the situational context of the news. Brussell retrieves news articles and creates and extracts situation models from templates. When a user selects a situation, it presents a storyline with the major milestone events. Clicking on the event label loads an article that either immediately covers the event or is the earliest mention of the event. Evidence that an event took place, for its date and location, or for important attributes of participating entities can also be viewed in the form of collected textual snippets and links to source pages.

2. EXAMPLE

Consider the case of a user reading about the history of the terrorist group Hamas. The article references the kidnapping of a BBC journalist, and although the user was vaguely aware of this incident, he would like to find out more. With standard search technology, he would enter terms into a search engine and peruse the results in order to develop an overall sense of how the kidnapping situation transpired. Through Brussell, he can interact with the textual reference directly, by first clicking on a button in the Brussell toolbar to show its situation reference "matches", then right-clicking on the highlighted text in the page (see Figure 1).

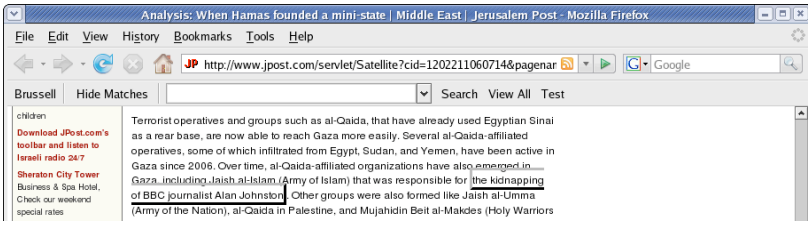


Figure 1. Viewing a situation reference within an article.

The context menu presents options for viewing the history of the situation and finding out more about its participants (see Figure 2). The user wants to see a summary of what happened, so he selects the first option, which updates the toolbar to show a storyline for the kidnapping with its major events and their dates (see Figure 3)

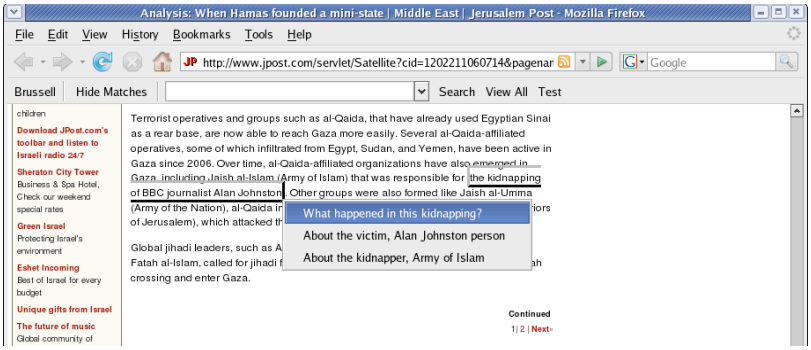


Figure 2. Asking about the situation.

Next, he wants to know more about how the journalist was released, so he selects the "release" event button that loads the most relevant page describing the event in detail (see Figure 4).

3. ARCHITECTURE

Brussell consists of a Firefox browser plugin and server software, which may both run on the same computer. When the user wants to inspect a situation reference the browser plugin sends the current page title and URL to the server, which responds with the (possibly cached) page situation references. A user can view situation references in news pages, as in the example, or can request the analysis of arbitrary web pages, such as blog posts.

Figure 3. Viewing milestone events for the selected situation

The back-end system requires manually-created situation model types (scripts) and currently supports *kidnappings*, *legal trials* and *corporate acquisitions* each of which has multiple possible outcomes and on the order of 8-12 possible events. The system runs daily to retrieve news articles from several news web sites via RSS feeds and store them in a Lucene index. [7] It then queries the database for new articles with keywords associated with the situation types it supports and reads through the returned articles to instantiate and extend situation models of these types. Situations include information from a few articles, up to several hundred if they are well-publicized.

Figure 4. Viewing the article for the selected situation event.

Brussell uses GATE [4], a standard open-source information extraction system to extract situation information including event references, dates and locations, and entity information such as person names and

occupations or organization names and nationalities. Extracting this information allows references such as "the British journalist abducted last year" to be resolved to a particular kidnapping. In fact, the same mechanism used for extracting information is used to identify situation references in page text, and in analyzing news articles, the system caches the textual references for all of articles it processes. Saving textual supports for extracted information serves an additional purpose: to justify how conflicting information has been reconciled.

3.1 Resolving Conflicting Article Information and Extraction Results

A well-known problem with building and manipulating explicitly represented models is that of resolving conflicting information. Often a breaking news article features incorrect information that is later amended. Or information in an article may be correct, but presented idiosyncratically and, as a result, extracted incorrectly. Based on the expectation that correct information will be stated more often than incorrect information, Brussell implements a voting algorithm to resolve error due either to incorrect article information or faulty extraction.

Voting is used to resolve conflicts at multiple levels:

- At the top-most level, to select which actual events occur within a situation
- Around event information including dates, locations and monetary amounts
- Concerning biographical information about situation participants such as person names and occupations or organization names and nationalities

A preliminary evaluation of this voting approach shows that the performance of relatively shallow extraction technologies integrated across multiple documents is comparable to more sophisticated extraction from single document, as found in, e.g., the MUC competitions.

4. BACKGROUND

Previous research has produced query-free information retrieval systems for end users such as Letizia [6] and Watson [3]. These systems search the web to find documents relevant to a user: Letizia by following the links of the currently open web page, and Watson by modeling her current task in the browser or an open Microsoft Office document.

Several areas of research have focused on distilling information from multiple news articles. Techniques in text summarization merge and reduce the information in multiple documents presenting the user with a natural language summary. [8] Research in topic-detection and tracking has focused on representing events, typically by term-vectors, and classifying and clustering documents using these event representations. [1] These domain-independent approaches do not model types of events and situations and the associated semantic constraints and thus cannot support users' expectations for the milestones of these situations and how they proceed. Our approach of modeling user expectations for situations is based on the script conceptual formalism for story understanding. [10].

Extracting event information using templates from single news articles was the focus of work in the Message Understanding Conferences [5]

One notable site that uses a model to extract and integrate information from multiple web pages is ZoomInfo.com, which automatically generates an individual's CV based on text references in web pages. [12]

5. FUTURE WORK

Two challenges remain for the system to scale not just on many articles, but many situation types. First, there is the problem of generating situation type models that consist of semantic constraints, document retrieval keywords and extraction patterns. Authoring the patterns is the most time-consuming component by far, though this could be automated through unsupervised learning techniques such as [9] or [13]

As more types of situations are modeled, support for richer knowledge representation will be required. For example, tracking an individual's employment at an organization would require representing an individual's occupation as multiple job records not just strings. Although trivial, it is expected that supporting more situation types will introduce many new representation requirements such as this one, each of which must be accommodated within the voting system.

6. CONCLUSION

Many researchers have put forward the goal of integrating the web with high-level semantic models to provide more goal-oriented interfaces. Some, including those working as part of the Semantic Web effort, expect to provide this user-level functionality by requiring authors to annotate their web pages using standardized domain-specific logical annotations. [2] In other words, this effort is aimed at providing smarter interactions with web content by constructing the web out of explicit logical representations.

We are taking the opposite approach to semantically-informed user interaction with web content. Rather than dragging the web to semantics, kicking and screaming, we are bringing semantics to the web. With Brussell, we have presented a system that enables users to interact directly with entities and situations mentioned in web pages in order to navigate the context of the content they are viewing. Brussell uses standard IR and IE technologies integrated with situation model templates to anticipate user questions, and provide links to - and summaries of - the answers resulting in high-level overviews of situations that match user expectations.

7. REFERENCES

[1] Allan, J., Carbonell, J., Doddington, G., Yamron, J., and Yang, Y.: 1998, 'Topic Detection and Tracking Pilot Study: Final Report'. In: Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop. San Francisco, CA, pp. 194-218, Morgan Kaufmann Publishers, Inc.

[2] Berners-Lee, T., Hendler, J. & Lassila, O. "The Semantic Web", Scientific American 284(5):34-43 (May 2001)

[3] Budzik, J. and Hammond, K. J. 2000. User interactions with everyday applications as context for just-in-time information access. In *Proceedings of the 5th international Conference on intelligent User interfaces* (New Orleans, Louisiana, United States, January 09 - 12, 2000). IUI '00. ACM, New York, NY, 44-51. DOI= http://doi.acm.org/10.1145/325737.325776

[4] Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V. GATE: A Framework and Graphical Development

Environment for Robust NLP Tools and Applications. *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*. Philadelphia, July 2002

[5] Grishman, R. 1997. Information Extraction: Techniques and Challenges. In *International Summer School on information Extraction: A Multidisciplinary Approach To An Emerging information Technology* M. T. Pazienza, Ed. Lecture Notes In Computer Science, vol. 1299. Springer-Verlag, London, 10-27.

[6] Lieberman, H., Letizia: 1995. An Agent That Assists Web Browsing, *Proceedings of the 1995 International Joint Conference on Artificial Intelligent*, Montreal, Canada, August 1995.

[7] http://lucene.apache.org/java/docs/

[8] McKeown, K. and Radev, D. R. 1995. Generating summaries of multiple news articles. In *Proceedings of the 18th Annual international ACM SIGIR Conference on Research and Development in information Retrieval* (Seattle, Washington, United States, July 09 - 13, 1995). E. A. Fox, P. Ingwersen, and R. Fidel, Eds. SIGIR '95. ACM, New York, NY, 74-82. DOI= http://doi.acm.org/10.1145/215206.215334

[9] Riloff, E. (1996) "Automatically Generating Extraction Patterns from Untagged Text", *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)* , 1996, pp. 1044-1049

[10] Schank, R. C. and Abelson, R. P. 1977. Scripts, Plans, Goals and Understanding. Lawrence Erlbaum Associates, Hillsdale, New Jersey.

[11] Sellen, A. J., Murphy, R., and Shaw, K. L. 2002. How knowledge workers use the web. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Changing Our World, Changing Ourselves* (Minneapolis, Minnesota, USA, April 20 - 25, 2002). CHI '02. ACM, New York, NY, 227-234. DOI= http://doi.acm.org/10.1145/503376.503418

[12] http://www.zoominfo.com/

[13] Yangarber, R. 2003. Counter-training in discovery of semantic patterns. In *Proceedings of the 41st Annual Meeting on Association For Computational Linguistics - Volume 1*

Summarization and Refinement Tags in Folksonomies

Joyce Wang
jwang@endeca.com

Vladimir Zelevinsky
vzelevinsky@endeca.com

Daniel Tunkelang
dt@endeca.com

Endeca
Cambridge, MA

ABSTRACT

Folksonomies improve search and navigation of documents by allowing users to collaboratively tag documents. Unfortunately, the number of tags can be overwhelming to users who are seeking information, even when the tags are restricted to those that occur in the search results. In this paper, we describe a novel approach for highlighting tags of interest for users, based on the premise that tags can be useful because they either summarize or refine the current set of results. We also present a treemap interface that visually communicates both kinds of tags to users. Finally, we present the results of a user study designed to test the validity of our approach.

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

folksonomies, summarization, refinement, treemap

1. INTRODUCTION

Folksonomies [1] are an increasingly popular way to enrich content and thus provide people with more effective ways to find information. In a folksonomy, a broad collection of people collaboratively tag documents. Folksonomies are also known as user-generated taxonomies.

One of the challenges in using tags to navigate a folksonomy is that the large number of tags quickly becomes overwhelming. In order to narrow the space of tags, we would like to highlight specific tags in order to help users both understand the data and find the tags that slice the data in interesting ways.

2. MEASURING THE UTILITY OF TAGS

We measure the utility of tags along two dimensions: how well a tag summarizes the information in a set of documents, and how well a tag refines that set into a useful subset. We consider two factors to inform a tag's inclusion in either of these sets: frequency with respect to the given set, and the distinctiveness of the subset of documents assigned that tag.

2.1 Tag Frequency

In a perfectly tagged collection, a tag would represent a perfect summary of a given set of documents if it were assigned to all of the documents in that set. Although folksonomies are not perfectly tagged, we hypothesize that a tag's effectiveness at summarizing a given set of documents is positively correlated to its frequency within the set.

It is harder to relate frequency to the utility of a tag as a refinement. What is clear is that the frequency should neither be too low, thus representing an insufficient fraction of the results, nor too high, thus not significantly narrowing from the given set.

2.2 Tag Distinctiveness

Given a collection of tagged documents, we compute the distinctiveness of a given set of documents relative to a baseline set by comparing the distribution of tags in the given set to that of the baseline. Specifically, we take a normalized Kullback-Leibler divergence (aka relative entropy, information gain). This normalization, which we accomplish by taking random subsets of the given set, is necessary to avoid confounding distinctiveness with set size, since smaller sets tend to have higher Kullback-Leibler divergence. This distinctiveness measure is inspired by Cronen-Townsend and Croft's "query clarity" measure [2].

As a short-hand, we refer to distinctiveness of a tag in a given set of documents as the distinctiveness of the subset of the given set that is assigned that tag, relative to the given set.

We now hypothesize that a tag with low distinctiveness will be useful for summarizing a given set. In particular, we conjecture that good summarization tag will have lower distinctiveness than good refinement tags.

3. VISUALIZATION

In order to simultaneously communicate the frequency and distinctiveness of tags, we implemented a tree map visualization. The tree map, a space-filling visualization technique developed by Ben Shneiderman, allows the visualization of two simultaneous attributes of a set of objects through the visual dimensions of cell size and color [3].

In our tree maps, the size of a cell corresponds to the frequency of the tag associated with that cell, while color corresponds to the position of the tag on the distinctiveness spectrum (darker being more distinctive and lighter being less distinctive).

Restating our earlier hypotheses in terms of the tree map, we expect that good summarization tags will correspond to large light-colored cells, while good refinement tags will correspond to medium-sized darker-colored cells.

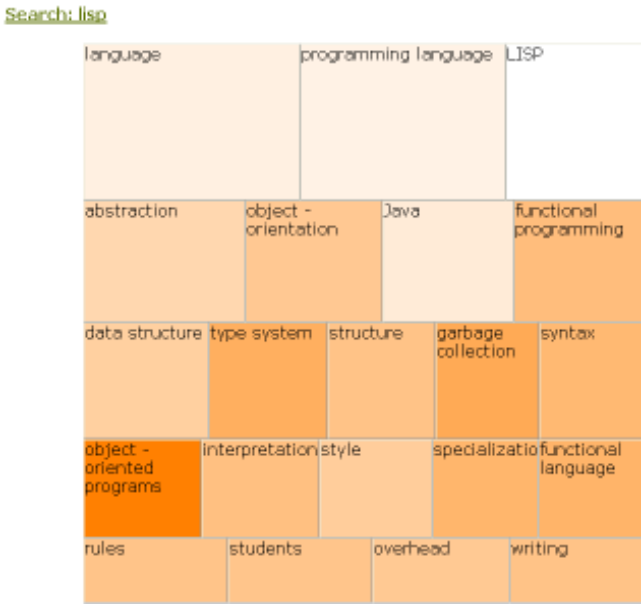


Figure 1: Tree map of a search for "lisp"

4. EVALUATION

We conducted a user study to empirically validate our hypotheses about frequency and distinctiveness determining the utility of tags for summarization and refinement. Specifically, the test was designed to explore whether subjective user judgments confirm those hypotheses. The user study also tested the effect of presenting users with the tree map visualization described above.

4.1 Experimental Setup

For our study, we used a subset of the ACM Digital Library which includes only author tagged documents. This data collection comprises over a quarter million articles, consisting of articles from ACM journals, conference proceedings, and newsletters [4].

In order to tag the corpus, we distilled a controlled vocabulary from the author tags assigned to the documents, keeping those with sufficient corpus frequency (assigned to at least 10 documents) and positive Residual IDF (RIDF) scores in accordance with a technique inspired by Church and Gale [5]. We then assigned tags to documents that contained the text of those tags (allowing for stemming) with sufficiently high TF-IDF scores. We note that this test set simulates a folksonomy by bootstrapping on a collective vocabulary, a technique we have applied in related work [6].

For each of 20 sets of ACM articles corresponding to search queries, we presented the user with two tasks: selecting the tags that best described the entire set, and selecting the tags that best described some of the articles (i.e., served as good refinements).

In the first task, we asked users to identify these two kinds of tags based on article titles and their author-selected keywords. In the second task, we asked users the same question, but instead showed them the search term that generated the set of articles and the tree map visualization described above.

To avoid ordering biases, we shuffled the displayed documents, and presented the list of possible tags in alphabetical order. Since we could not display all of the available tags without overwhelming users, we showed those tags that occurred in at least 3.5% of the documents in the set. In the first task, we further limited the number of tags to 20 if needed (the 20 most frequent) in order to avoid presenting the user with too much information. There was no such limitation on the number of tags in the second task, where we presented the user with the tree map visualization.

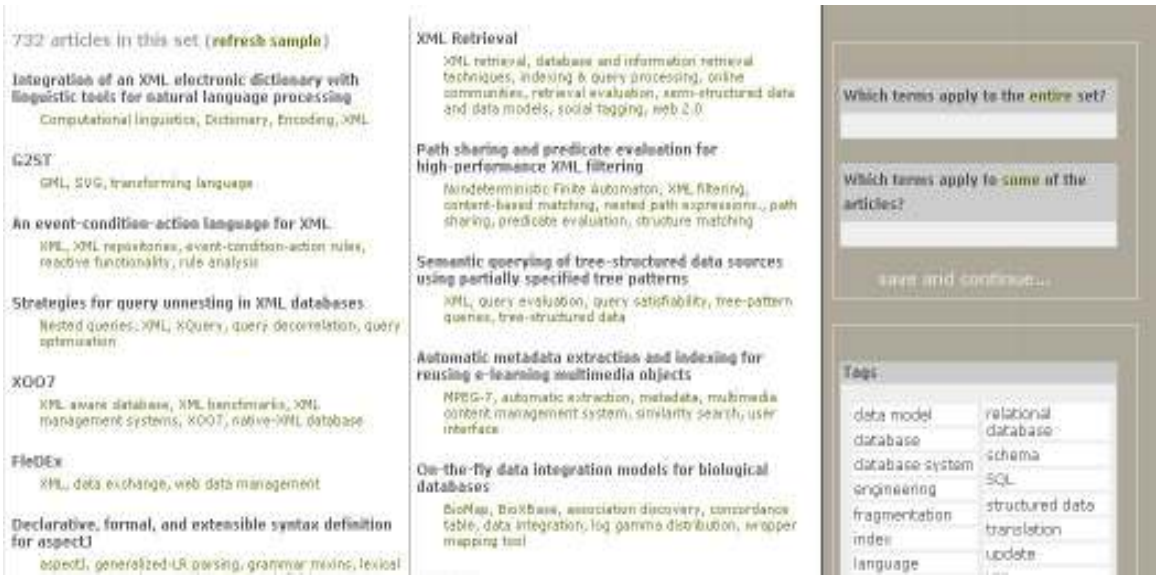


Figure 2: User study tasks

We also gave the user the option of displaying more documents from the given set (effectively paging through the shuffled ordering), as well as the option of viewing the abstract of a specific document, rather than just its title (Figure 2).

We note that there were no "right answers" for the test queries, since users were making their own judgments regarding how well tags summarized or refined the sets of documents. Rather, we were using their subjective judgments as ground truth.

4.2 Hypotheses

We now formalize the hypotheses our user study aimed to validate regarding relationships between tag frequency, tag distinctiveness, utility for summarization, and utility for refinement:

1. Good summarization tags have high frequency.
2. Good summarization tags have low distinctiveness.
3. Good summarization tags have lower distinctiveness than good refinement tags.
4. Users' accuracy and efficiency in identify the tags with the highest utility for summarization and refinement will increase when presented with a tree map visualization of frequency and distinctiveness.

5. RESULTS

We had 36 total participants in the user study, all with at least a bachelor's degree in computer science or comparable background. 24 of the participants completed the roughly one-hour user study.

For each set of articles, each user response consists of an unordered set of tags that the user found most suitable to 1) describe the entire set ("summarize"), and 2) describe some of the articles in the set ("refine"). Aggregating these responses gave us the number of times a particular tag was chosen for the set. Each of these tags has a frequency and a distinctiveness score associated with it.

To analyze the results of our user study, we took the averages of the frequency and distinctiveness scores in the user responses for the first task. We used as our baseline the average frequency and distinctiveness scores for all tags displayed to the user in a given set. Table 1 show example scores for three of the 20 test queries.

Query	xquery	scrum	backgammon
Baseline Frequency.	0.126	0.113	0.112
Summarize Frequency	0.364	0.115	0.114
Refine Frequency	0.150	0.101	0.126
Baseline Distinctiveness	5.496	7.382	9.540
Summarize Distinctiveness	3.838	7.385	6.811
Refine Distinctiveness	4.985	7.489	8.138

Table 1: Scores for Selected Queries

One-tailed t-tests show statistically significant results at the 0.05 level for the following hypotheses:

- Frequency of user-selected summarization tags > baseline frequency.
- Distinctiveness of user-selected summarization tags < baseline distinctiveness.
- Distinctiveness of user-selected summarization tags < refinement distinctiveness.

These tests support our first three hypotheses; that is, good summarization tags have high frequency and low distinctiveness, and in particular lower distinctiveness than good refinement tags.

Unfortunately, we were not able to establish useful criteria to distinguish between good refinement tags and the baseline based on frequency and distinctiveness, other than their not being good summarization tags. We did find that refinement frequency was higher than baseline frequency (statistically significant at the 0.05 level), but all we can infer from this result is the obvious fact that good refinement tags should not be too infrequent.

Finally, we were not able to draw quantitative conclusions from our second task to validate our fourth hypothesis. As we realized from post-study discussions with our participants, it was impossible to present the visualization without those participants trying to reverse engineer what it meant.

6. CONCLUSION

Our user study validated our basic hypotheses regarding relationships between tag frequency, tag distinctiveness, utility for summarization, and utility for refinement. We hope to follow up this experiment with a larger-scale study that uses ground truth data (e.g., from trained assessors) to establish summarization and refinement utility.

7. REFERENCES

[1] Vanderwal, T. (2005). Off the Top: Folksonomy Entries. http://www.vanderwal.net/random/category.php?cat=153

[2] Cronen-Townsend, S. and Croft, W.B. 2002 Quantifying query ambiguity. *Proceedings of the Second International Conference on Human Language Technology Research* (March 2002), 104-109.

[3] Shneiderman, B. (1991). Tree visualization with treemaps: a 2-d space-filling approach. *ACM Transactions of Graphics*, vol 11, 1 (January 1992), 92-99.

[4] ACM Portal: http://portal.acm.org/

[5] Church, K. and Gale, W. (1995). Inverse Document Frequency (IDF): A Measure of Deviation from Poisson. In *Proceedings of the Third Workshop on Very Large Corpora*, 121-130.

[6] Zelevinsky, V., Wang, J., and Tunkelang, D. (2008). Supporting Exploratory Search for the ACM Digital Library. Submitted to *Second Workshop on Human Computer Information Retrieval (HCIR '08)*.

Site Metadata on the Web

Erik Wilde
School of Information
UC Berkeley
dret@berkeley.edu

ABSTRACT

The navigation structure of Web sites can be regarded as metadata that can be used for interesting applications in *User Interface (UI)* design and *Human-Computer Interaction (HCI)*, as well as for *Information Retrieval (IR)* tasks. However, there currently is no established format for site metadata, which makes it hard for Web sites to publish their structure in a machine-readable way, which could then be used by HCI and/or IR applications. We propose a model and a format for site metadata that is built on top of an existing format and thus could be deployed with little overhead by publishers as well as consumers. Making site metadata available as machine-readable data can be used for improving user interfaces (informing user agents about the context of the page they are displaying) and better information retrieval (allowing search engines to use sitemap information for better ranking and display of the results).

Categories and Subject Descriptors

H.5.4 [**Information Interfaces and Presentation**]: Hypertext/Hypermedia—*Navigation*

General Terms

Design, Standardization

1. INTRODUCTION

The URI structure of a Web site (often referred to as a *site map*) is an important aid for navigating the content of a site. Many Web sites make the site structure available through *site navigation*, often implemented visually as horizontal and/or vertical menu bars, or less frequently also through a dedicated Web page representing the site map, listing all of the site's available pages. However, there currently is no machine-readable format for this information, which we call "site metadata." This paper discusses the challenges and the potential benefits of such a format, and proposes a way to augment the *sitemaps.org* format with site metadata.

Site Metadata on the one hand greatly improves the interaction of humans with a site, because many tasks on a site require accessing more than one page on the site. On the other hand, even though explicit navigation often is provided through Web page design, IR can be used to algorithmically infer site metadata for tasks other than direct user interaction with a Web site. Google's search results, for example, occasionally include a small "site map" (called "sitelinks") for highly ranked search results (Figure 1 shows an example). Allowing Web sites to publish site map data in a machine-readable way thus could augment HCI as well as IR tasks regarding Web page structures.

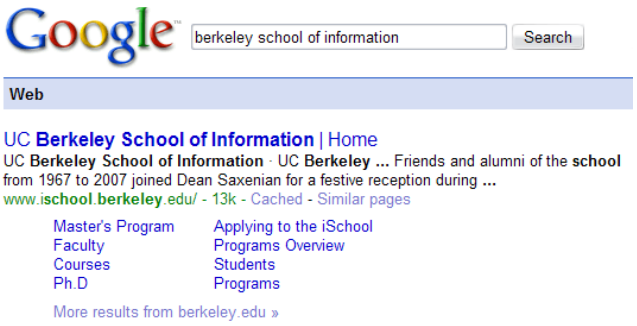


Figure 1: Algorithmically Determined Sitelinks

Sections 2 and 3 give a short overview of the possible benefits of explicit site metadata on the Web, and Section 4 summarizes this potential. Section 5 then describes the data model that we have defined so far, and Section 6 then makes a proposal for augmenting an already existing format with site metadata based on that model.

2. NAVIGATION SUPPORT FOR HUMANS

While usability and accessibility are important subjects in the context of individual Web pages, usability and accessibility of Web sites (i.e., a structured and interconnected set of Web pages) is a topic that is discussed less frequently. HTML itself has the ability to include `<link>` elements in the document head which can express a number of document relationships between HTML documents, but the available relationship types indicate that the focus of this feature is to support single logical documents which are represented by more than one HTML document. Furthermore, most browsers do not support this HTML feature.[1] And since it is defined in HTML itself, it cannot be used easily to cover HTML as well as non-HTML media types.

[1]Only Opera natively support navigating `<link>` elements; for Firefox and IE there are extensions supporting this functionality.

Second Workshop on
Human-Computer Interaction and Information Retrieval (HCIR 2008)
October 23, 2008, Redmond, WA

The *Web Content Accessibility Guidelines (WCAG)* [1] also do not discuss in great detail how to make the navigational structure of Web sites accessible, they mainly focus on making document structures accessible. WCAG technique G62 recommends to provide a site map, but talks of that site map as an HTML page, which means that the sitemap is not machine-understandable.

On today's Web, the navigational structure of a Web site is usually represented visually by common "design patterns" for Web-based user interfaces, and in most cases the actual data is provided by a *Content Management System (CMS)* on the back end, which propagates the design pattern with site data.[2] Even though there is a small number of these design patterns describing the vast majority of Web sites, this still leaves navigational structures in the realm of Web information not described in a machine-understandable way.

There is only little research about how better orientation within a Web site could help users to better navigate and utilize the site. One study conducted by Danielson [3] suggests that constantly visible site maps do have a positive effect on how people can utilize a site in terms of more effective navigation and a better overview of the available resources on a site.

3. SITE METADATA FOR MACHINES

The *sitemaps.org* format has been invented by Google and now is being jointly developed by a number of major search engines. Despite its name it is not a site map, it is simply a set of URIs which can be provided by Web masters to provide search engine crawlers with a set of URIs they might want to crawl. The intent of the sitemaps.org format is not to provide information about a site's structure, but only to provide information about the accessible URIs.

In addition to the basic text format (a list of URIs, one per line), there also is an XML format. This format allows Web masters to specify additional information for individual resources, the last modified date, the expected change frequency, and a priority. Crawlers are free in how they use that information to control the crawling process, and most crawlers will use internal heuristics to decide how much they rely on this additional information.

4. POTENTIAL OF SITE METADATA

While the goals of using site metadata for supporting humans (Section 2) or machines (Section 3) are different, both goals could be accomplished by using the same metadata. The following list is likely to be incomplete, but lists some of the areas where site metadata could be used to provide better implementations of HCI- or IR-related tasks.

- *Unified Navigation:* If site metadata were available to browsers, they could provide unified controls for navigating sites, making it unnecessary for users to adjust to the various ways in which sites implement site navigation.[3] Browser navigation not necessarily has to completely replace the embedded navigation, but a browser could provide additional features to better guide users through a site.

- *Accessibility:* Even though Web page accessibility is a popular topic, this is much less true for Web site accessibility, i.e. the ability for users to navigate a Web site without having to search through embedded navigation controls. Site metadata can greatly improve site accessibility, because it allows browsers to explicitly provide navigation features, without the need to "find" the embedded navigation controls of Web pages.

- *Crawling:* The sitemaps.org format already has most important information that allows crawlers to adjust their strategy to a site's resources. However, more navigational data (such as the various "levels of hierarchy" on a Web site) might also be useful input for determining crawl sequences.

- *Ranking:* Based on a site's structure, ranking can be better informed because hits could be ranked according to specificity (a hit in a page "lower" in the hierarchy is likely to be more specific, whereas a hit in a "higher" page is more likely to be on an overview page). As for crawling, ranking could use this information as additional input to already existing strategies and algorithms.

- *Search Result Clustering:* In a way similar to that shown in Figure 1, site metadata could be use to cluster search results according to a site's structure, or to show where in a site's structure a hit occurred. Again, site metadata would most likely only be one input into such a feature.

While the HCI-oriented tasks (unified navigation and accessibility) make use of the site metadata on a per-site basis, the IR-oriented tasks are based on using the aggregated site metadata of a large number of sites. As usual, Web masters might be tempted to try to game algorithms by supplying site metadata that should improve a sites visibility in a search engine. Site metadata in such a scenario might become just one more factor in what is often referred to as *Search Engine Optimization (SEO)*, which comprises a number of legitimate and useful ways to improve a sites usability for search engines, but sometimes also includes strategies which run against the intentions of search engine providers and have to be detected and compensated for.

Machine support by site metadata is already partially supported by the sitemaps.org format, but there is only very little support for site navigation for humans. One notable exception is the *Standard-Navigation* (formerly known as *Standard-Sitemap*) Firefox add-on shown in Figure 2. It uses a custom XML format which supporting Web sites are supposed to supply, and then uses that data in a browser sidebar. The add-on even has the option to hide the embedded navigation on a Web page (which has to be marked up with specific HTML code), so that navigation controls will only be displayed in the sidebar, and not also as embedded controls in the Web page.[4]

[2]The *Web Modeling Language (WebML)* [2] supports an elaborate model of how to describe datasets and Web interfaces for them.

[3]In a simple way this already is possible if a site uses a well-design URI structure, where the navigation hierarchy is reflected in the URI hierarchy. In this case, simple browser extensions such as the Firefox *Go Up* extension allow users to go up one level on the site by using a browser button.

[4]Browsers not using the add-on will not recognize the special markup for the embedded navigation controls and will therefore not hide them.

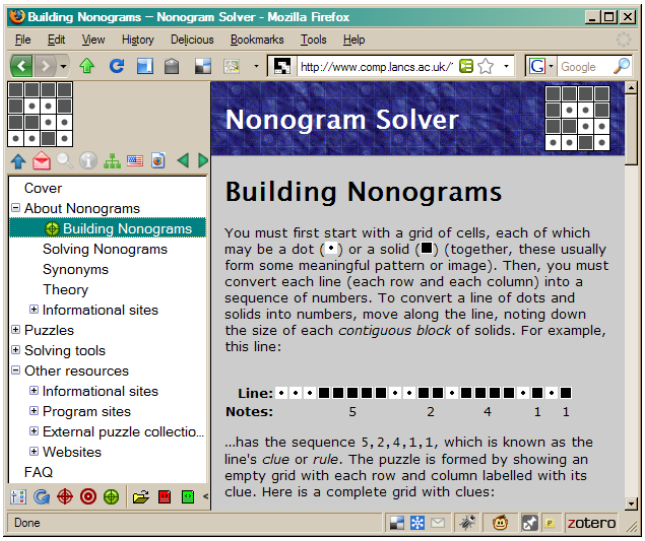


Figure 2: Standard-Navigation Sidebar

The approach of this add-on is to completely remove embedded navigation from Web pages, so that all navigation can be controlled through the sidebar. It is at least questionable whether this is a goal that will be shared by a substantial share of Web designers. We believe that it's more useful to think of browser-based controls for navigation as supplemental features for whatever the Web designers choose to embed within their Web pages. It then remains to be seen (and tested) how useful a more unified towards navigation actually is, and how much there will be a general trend towards outsourcing navigation controls from Web page content to browser controls.

5. SITE METADATA DESIGN

At first sight, the design of a site metadata model might seem almost trivial. A simple sitemap usually can be modeled as a tree representing the hierarchical structure of a Web site. For very simple sites, this model might be complete or at least sufficient, but when looking at Web sites, it quickly becomes apparent that site metadata can be much more complex in structure than just a simple tree with one kind of relation between resources. The following issues illustrate some of the potential complications of real-world site metadata:

- *Sets vs. Sequences:* While some sites might want to model their hierarchical structures as sets, other might want to model them as sequences. Moreover, in the case of sequences, the actual sequence can sometimes depend on factors which vary with resource variants (such as page titles, which will vary by language).
- *Variants:* Resources (navigation targets in the site structure) might exist in different variants, and the variants might use different dimensions of variation. Typical examples are languages (multilingual Web sites) and media types (resources might be available as HTML and PDF). While all of these resources are equivalent on a conceptual level, concrete clients will most likely only use one of them, depending on user preferences and client capabilities.
- *Versioning:* Versions can be regarded as a special type of variant because they have the built-in assumption that there is a chronological sequence of versions. Complex version models might be non-linear, for example when a page is split into multiple pages and thus the versioning structure becomes a tree (in general, versioning graphs are directed acyclic graphs).
- *Non-Tree Structures:* While many sites indeed are tree structured, there are also sites where the navigation structure "reuses" pages in various locations, so that the effective navigation structure can either be regarded as a tree with duplicate pages in it, or as a directed acyclic graph.
- *Dynamic Structures:* Advanced Web sites sometimes customize navigation structures based on criteria such as a personal profile, histories, preferences, and popularity of pages with recent visitors. With these sites, site metadata is determined by many different factors and the navigation aspects of site metadata have to be specifically determined for each client. However, there is no reason why the dynamic generation of embedded navigation controls could not also drive the generation of site metadata.
- *URI-less Navigation:* While many sites do have individual URIs for different pages in their navigation structure, there are also sites which do not have URIs for these pages. The two most common cases for this are frame-based sites, and sites where embedded code (popular examples are Ajax and Flash) handles navigation without reloading pages.

The above list of issues probably supports the way in which most Web sites would want to publish their site metadata, but it might also exclude some sites which have even more sophisticated models of their site's structure. Also, because the *Hypertext Transfer Protocol (HTTP)* [4] provides functionality beyond the simply retrieval of resources, some of the complexity of the above list could be deferred to HTTP.

For example, the detection of variants could be deferred to HTTP content negotiation, which allows Web servers to advertise that a resource is available in different variants. But many Web sites do not use HTTP-based language selection, they simply provide different resources without any machine-readable information about their conceptual equivalence. If a site metadata model should also support these sites, then variants must be included in the model.

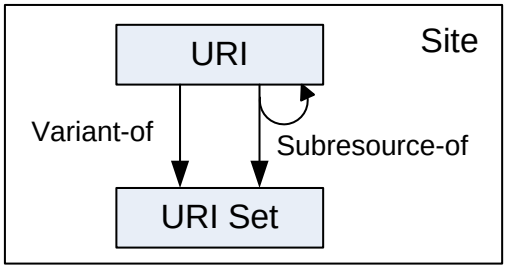


Figure 3: Site Metadata Model

Based on these considerations, we have designed the site metadata model shown in Figure 3. We decided to not

include versioning information, because it complicates the data model, and there were only few use cases where version information was a required component of the data model.

A site is escribed by a number of URIs and URI sets. possible relationships between URIs are hierarchy levels (expressed by the *subresource-of* relation), and if a resource is represented by multiple variants, a URI set is used (associated by the *variant-of* relation). URIs are associated with URI sets by specifying the dimension(s) of variation and the respective value(s). Optionally, URIs and URI sets can have position values, which are used to determine a sequence of resources, if sites want to use sequences rather than sets.

6. DATA FORMAT

The data model for site metadata described in Section 5 can be represented in different ways. We identified the following three methods as the most promising candidates for representing site metadata:

- *Dedicated XML Format:* It is possible to create a entirely new data format, and XML is a good choice because it has become the most widely supported foundation for the open exchange of structured data.
- *RDF:* Since site metadata is not content but metadata about content, it might be regarded as something that should be represented using *Semantic Web* [5] technologies, using the *Resource Description Framework (RDF)* as its model and syntax.
- *Extension of existing XML Format:* Instead of starting from scratch, an existing format could be extended. The most promising candidate is the sitemaps.org format.

We decided that the most promising way is to extend the sitemaps.org format, which seems to have gained some popularity (even though we could not find any data about that). Unfortunately, the extensibility (as well as the format as a whole) is very poorly documented, which makes it impossible to understand what kind of extensions the format allows. This is relevant because existing implementations might break or misinterpret data if they have built-in assumptions about the data format which have not been documented in the format itself, and which are violated by an extension.[5]

Based on the limited information about extensibility, the current format could be updated as follows: URI sets are represented by the `urlset` element, which is allowed as a child of the `urlset` document element. The `url` and `urlset` elements have an optional `id` attribute, and a subresource is identified by an `parent` attribute which specified the ID of the higher-level resource. Optionally, a subresource can carry a `pos`ition attribute for specifying a sequence of subresources rather than a set. Variants use a `variant` element as a child of the `url` element, and this element has attributes for the `urlset` (it is a variant of this URI set), the `dim`ension (such as language or media type), and the `value` for that dimension (such as a concrete language).

[5]Google claims that a well-defined extensibility model is under development, but in contrast to the data model, which is openly available and CC-licensed, the development process is closed and no information about the extensibility model is currently available.

```
<urlset xmlns="http://www.sitemaps.org/xmlns/1">
  <url id="home">
    <loc>http://www.example.com/</loc>
  </url>
  <url id="contact" parent="home" pos="1">
    <loc>http://www.example.com/contact</loc>
  </url>
  <urlset id="faq" parent="home" pos="2"/>
  <url>
    <loc>http://www.example.com/faq,en</loc>
    <variant urlset="faq" dim="lang" value="en"/>
  </url>
  <url>
    <loc>http://www.example.com/faq,de</loc>
    <variant urlset="faq" dim="lang" value="de"/>
  </url>
</urlset>
```

While the main structure of the sitemaps.org format remains the same, the addition of attributes and a new child element type to the document element might be something that is considered out of scope for extensions. If that is the case, the above example can also be represented using only new child elements of the `url` element. This kind of representation is even more verbose and less elegant, but the most important issue is that the data model (Section 5) can be represented in an extension of the sitemaps.org syntax.

7. CONCLUSIONS

In this paper, we present our work towards making site metadata available on the Web. The current sitemaps.org format has gained some popularity and is useful for the IR-oriented tasks regarding site metadata, but it ignores the benefits that are possible from an HCI perspective towards better site navigation for users. Our future work is twofold: When the revised sitemaps.org format is released, we will have a well-defined set of rules for this data format. On the other hand, we want to explore the possibilities and limitations of navigation support driven by site metadata. This exploration of the usefulness of site metadata will inform our final definition of the sitemaps.org extension; it is of course possible that our current data model will have to be revised.

8. REFERENCES

[1] Ben Caldwell, Michael Cooper, Loretta Guarino Reid, and Gregg Vanderheiden. Web Content Accessibility Guidelines 2.0. World Wide Web Consortium, Candidate Recommendation CR-WCAG20-20080430, April 2008.

[2] Stefano Ceri, Piero Fraternali, and Maristella Matera. Conceptual Modeling of Data-Intensive Web Applications. *IEEE Internet Computing*, 6(4):20–30, 2002.

[3] David R. Danielson. Web Navigation and the Behavioral Effects of Constantly Visible Site Maps. *Interacting with Computers*, 14(5):601–618, October 2002.

[4] Roy Thomas Fielding, Jim Gettys, Jeffrey C. Mogul, Henrik Frystyk Nielsen, Larry Masinter, Paul J. Leach, and Tim Berners-Lee. Hypertext Transfer Protocol — HTTP/1.1. Internet RFC 2616, June 1999.

[5] Nigel Shadbolt, Tim Berners-Lee, and Wendy Hall. The Semantic Web Revisited. *IEEE Intelligent Systems*, 21(3):96–101, March 2006.

Improving Exploratory Search Interfaces: Adding Value or Information Overload?

Max L. Wilson, m.c. schraefel
School of Electronics and Computer Science
University of Southampton, UK
{mlw05r, mc}@ecs.soton.ac.uk

ABSTRACT
One method for supporting more exploratory forms of search has been to include a compound of new interface features, such as facets, previews, collection points, synchronous communication, and note-taking spaces, within a single search interface. One side effect, however, is that some compounds can be confusing, rather than supportive during search. Faceted browsing, for example, conveys domain terminology and supports rich interaction, but can potentially present an abundance of information. In this paper we focus on the faceted example and conclude with our position that Cognitive Load Theory can be used to estimate and thus manage the potential complexities of adding new features to search interfaces.

INTRODUCTION
The recent interest in supporting more exploratory forms of search [13], for when users are unfamiliar with domain terminology, information sources, or even their own goals, has spurred many new interface design ideas. One method that mSpace, Figure 1, has promoted for supporting a range of directed and exploratory search behaviours, has been to provide a gestalt of interface features [9]. Similarly, the latest version of the Relation Browser has recently extended their range of visualisations and interactions, including the addition of facet clouds [2]. Further, the recent Parallax interface to the Freebase project[1] provides a combination of faceted search, fact views, timelines, and maps to help users explore a wide range of heterogeneous data.

Both the mSpace and Relation Browser interfaces, and many others, provide a user interface with a compound of features, where the aim is for the set of features to work together in synergy in supporting users during search. Conversely, however, Schwartz has discussed the paradox of choice in that often, when users are presented with increasing numbers of options, they make poor or possibly no decisions [10]. In line with Schwartz's findings, many online faceted search websites focus on reducing decision paralysis by presenting only the key facets and their key options at each stage of the user's search [11]. This is most notable when facets, such as those presented by eBay start with a small set of values with a link to see 'more' options.

[1] http://mqlx.com/~david/parallax/ - Freebase Parallax

Figure 1: mSpace is a Directional Consistent Faceted Browser.

Evidently, there are two opposing forces that will affect the design of future exploratory search interfaces: 1) enriched functionality and 2) clarity in design. Unfortunately, recent work has also described the difficulties that can be faced when trying to evaluate the proposed advances in exploratory search interfaces [14].

In the next section of this paper we focus this problem by assessing the different approaches taken in providing one type of exploratory search feature: faceted browsing. We identify two dimensions that are present in the different implementations of faceted search and detail both the arguments for and against them. In the latter half of the paper, we propose that Cognitive Load Theory (CLT) [3] can be used to estimate the severity of the expected costs of the different approaches to faceted browsing. Further, the theory can be included into an existing, validated inspection framework [16] so that designs are evaluated for both synergy of features *and* complexity of design.

DEFINING THE DIMENSIONS OF AN EXAMPLE EXPLORATORY FEATURE: FACETED BROWSING
Faceted browsing [5] is an approach to supporting exploratory search that takes a set of meta-data from a corpus and presents the different attributes, and the distinct set of instances from each attribute, to the user. When shopping online for dresses, for example, users may make selections within facets such as price, colour, size, style,

and material, to reduce the number of purchasable items. In general, faceted browsing has a number of expected benefits over typical keyword search [5]. One example is that faceted browsing provides users with options to choose from when searching, so that they do not have to guess keyword search terms on their own.

Although various faceted browsers are unified in their aim to provide these expected benefits to exploratory users, there is significant variation in their implementations. In particular, there are two main dimensions that vary in faceted browsers: 1) direction between facets and 2) consistency of display. These dimensions are discussed so that later, their costs can be more concisely understood, explained with CLT, and managed in the future.

Dimension 1: Direction between Facets

Apple's iTunes is an example of a faceted browser that maintains direction between facets. Selecting an Artist filters the list of Albums, but not the Genre column. Like iTunes, most directional faceted browsers present facets in a series of columns across the interface from left to right. mSpace is a directional column browser that has overcome the problem that no Genre associations are shown [15]. Most other instances of faceted browsing, like those on Google product search, Walmart, and eBay, present facets that are unanimously filtered by any selections. Selecting a price range in Google Product Search filters every facet regardless of location of facets on the screen.

The perceived benefit of keeping direction is that additional relationships between facets are clearly shown. In iTunes, selecting a Genre will filter both the Artists and the Albums. Choosing an Artist then filters the Albums, but not the Genres. Now the user sees all the Artists in the selected Genre and all the Albums from the selected Artist. One perceived "'problem'" with maintaining direction is that it can overload the users, as they would have to maintain both a notion of direction, understand the relationships between side-by-side facets, and choose which facet and value to select next to refine their search.

Dimension 2: Consistency of Display

One hypothesis, held by browsers such as Flamenco [17], is that hiding used facets and dedicating screen space to unused facets can minimize information overload. Similarly, browsers often default to show the only the most popular values in a facet to reduce the number of choices. As previous decisions, and their options, are hidden using this method, previous choices are usually placed together as a breadcrumb trail. Another benefit of this approach is that once a user's decision has been hidden, the space can be given to show sub-category options of that selection.

One potential problem with hiding used facets and making space for unmade decisions is that it can be hard to quickly compare multiple items within one facet. In order to compare one style of dress with another, users are required to make an extra step to undo their first action, before making another selection. Further, by hiding used facets, it becomes difficult for a user to make multiple selections within one facet and see the dresses in two or more styles.

The intersection of these Dimensions in Browsers

These two dimensions produce a grid, as shown in Table 1. As noted before, iTunes and mSpace are the two notable examples of faceted browsers that choose to have a direction between facets that affects which are filtered by a selection. Combined with the choice of a consistent layout, these browsers provide: a) inter-facet relationships, b) multiple selections in any facet, c) previous decisions, d) previous selections e) all unused facets and f) a result set.

The remaining browsers listed in Table 1 are all examples that do not employ a direction but allow any facet to be filtered by the facet, and value, chosen by the user. Of these remaining browsers, most also chose to hide the used facets as the users make decisions (Varying layout). As a result, the user neither has to worry about the concept of a direction can choose freely among the facets and only has to consider the facets that remain in view. This combination, however, only provides: a) previous selections b) all unused facets and c) a result set.

Table 1: Examples of Faceted Browsers categorised by Use of Direction and Consistency of Layout

	Consistent Layout	**Varying Layout**
Directional Filtering	**e.g.** mSpace, iTunes.	?
Universal Filtering	Exhibit, Relation Browser	Flamenco, eBay, Endeca, Google.

Exhibit is an example of a non-directional, but consistently laid out faceted browser, where used facets are not hidden. This means that the inter-facet relationships from the Genre/Artist/Album iTunes scenario can be created by the order of selections, as opposed to the order of the layout. Although this approach produces the same result set and values in each facet as a directional and consistent browser, there is yet no evidence to show that the unstructured layout makes the relationships as clear as having the three facets side-by-side. In summary, this approach provides: a) multiple selections in one facet, b) previous decisions, c) previous selections, d) all unused facets, and e) a result set.

It is worth noting here that no browser has yet attempted to provide direction in their filtering, whilst hiding previous decisions to make space for unused facets. This maybe because hiding previous decisions also removes the ability to see the inter-column relationships provided by directional browsing. Further, the combination would hide potentially unused facets (in the iTunes problem, selecting an Artist would put both the Artist and the Genre column out of view). This combination would appear to provide only a) previous selections and b) a result set.

THE COSTS ASSOCIATED WITH THESE DIMENSIONS

While the previous section indicates that some browsers have potential functional benefits over others, the opposing argument is that each additional benefit comes at a cost of interface complexity provided to the user. In the directional and consistently laid out browsers like mSpace and iTunes, the user has to comprehend the effect of direction and consider both facet-result and facet-facet relationships.

Consequently, we are left with the challenge of trying to estimate which approaches are '*better*' for the user. Certainly, the majority of examples of faceted browsers on the Web choose the less complicated non-directional and space-optimising layouts, which we consider to have less functional benefit. Alternatively, iTunes has chosen the more powerful, but perhaps more challenging approach of providing a directional and consistent layout. Wilson *et al.* have already produced an inspection-based evaluation framework that can analyse the extent of functional benefits provided by search interfaces, but consequently encourages the complicated directional and consistent designs provided by mSpace and iTunes [16]. We now discuss Cognitive Load Theory, which we believe can be integrated into the same framework to argue against complexity. The extended framework would support designers in deciding if the added benefits of new features outweigh the added complexities.

Understanding the costs using Cognitive Load Theory

Put simply, the notion of Cognitive Load Theory (CLT) is that the complexity of a learning task and any learning material both affect the users ability to gain the knowledge they seek [3]. The complexity of a learning task is called *intrinsic load*, and learning materials should aim to support users no matter how much intrinsic load their task requires. If a problem is too big for working memory, then learning material should support users in breaking it down into steps, each with lower intrinsic load. Learning materials, or the objects that support users in learning, provide *extraneous load*. The aim of learning material should also be to reduce its extraneous load on the user, so that more intrinsically loaded tasks can still be achieved. If the extraneous load is high, then only tasks with a low intrinsic load may be achieved. Ultimately, however, both need to be reduced to make space in the overall cognitive load, for *germane load*, which is required to commit anything learnt into schemas in long-term memory. According to CLT, although space for germane load can be produced by minimizing intrinsic and extraneous load, the design of learning materials can effect whether or not the space is used for germane load.

So far, CLT has been designed to understand how instruction manuals, for example, can be better designed to teach people to use machinery or computers [4]. In these scenarios, the task has been to learn how to use a computer and the material has been a book. Learning, however, is often the same task held by exploratory search users, except that the material they have to support them in achieving their goal is a search interface. Ultimately, the user is still aiming to learn something, and has resources to help them do it, and so our first position in this paper is that CLT can be applied to understand the complexity of search software. This position supported by Mu [7], who, states 'cognitive loads are closely related to the complexity of a task, the system used to operate the task, and the operators characteristics', which makes no indication that 'the system' need be instructional. Further, others have considered how CLT might help interface designers convey search result relevance [6] and explain why users rarely provide relevance feedback during search [1].

The next stage is to translate the methods that CLT has identified for reducing the complexity of instructional material, to the reduction of complexity in search interfaces. CLT presents three methods of improving instructional material: split-attention, modality, and redundancy effects.

Split Attention Effect refers to occasions when a user has to mentally integrate information from multiple sources, such as text and a diagram, in order complete their learning. Chandler and Sweller approach this problem by making sure that the text necessary to understand a diagram is embedded within the diagram [4]. Otherwise, the system places unnecessary extraneous load on users, as they have to remember textual information while interpreting the diagram, or visa versa. An example here, from mSpace, may be that previous choices are highlighted and left in place, rather than displayed as a separate list of choices in a separate location [15]. Consequently, users can see both their decision and choices in place. Conversely, it may be better to have all your choices in one breadcrumb-style place, rather than having to find them in multiple facets.

Modality Effect refers to the reduction of cognitive load, by distributing learning into the different modalities of working memory. mSpace has tried this with audio preview cues so that users may take advantage of the auditory channel when making decisions about musical domains [8]. Similarly, the Relation Browser provides graphical volume representations with each facet value, which uses a separate mode to numeric values [18].

Redundancy Effect refers to situations where the same information is displayed in multiple places, so that the user is potentially required to a) read information they have already read and b) recognize what is new or has already been seen. Chandler and Sweller further their previous diagram and text example, by removing text that simply states what is clearly demonstrated by the diagram. It would appear, for example, that reducing the redundancy effect might help protect users from decision paralysis [10].

Using CLT within an Inspection Evaluation Framework To Manage and Reduce these Costs

Most research into CLT measurement has focused on recording the actual experience of users, through physiological changes, subjective views, task performance, and secondary-task performance (where their ability to multi-task is reduced by high cognitive load). An inspection

framework, however, focuses on assessment through careful estimation by some model and expected metric. Very little has been written about how to formally estimate cognitive load, but Chandler and Sweller [4] provide the following guidelines for estimating element interactivity: 'the extent to which elements interact for any given instructional material may be estimated a priori by simply counting the number of elements that must be considered simultaneously in order to learn a particular procedure.'

This process can be easily integrated into the authors' inspection framework [16], as it already counts the users 'moves' required to achieve a task. Chandler and Sweller add a caveat that this can only be applied in consideration of the user's existing capabilities. As the inspection evaluation framework also has a model of user types, this should also be easy to integrate. Further, as the framework already calculates the different interface features that allow users to carry out the same strategy, then we can also integrate measures for *split-attention* and *redundancy*.

With CLT integrated into the inspection framework, results would allow assessors to easily compare the extraneous loads produced by, in our example, different faceted browsers. This may first tell us if there is any significant cognitive load difference between the various approaches. Second, the framework would allow assessors to compare the difference between the increase in search support provided by each interface feature and the extraneous load produced. Third, the nature of the framework would allow assessors to quickly, and incrementally, consider design changes for both enriched support and reduced cognitive load. Having such a measure would complement cognitive engineering guidelines, such as the Ecological Interface Design framework [12], which encourage designs that require lower amounts of working memory.

CONCLUSIONS AND FUTURE WORK

In this paper we address the problem of a) finding the best trade-off between rich functionality and clear design, and b) discovering which combination of features best supports exploratory search. Using the inherent variation found in faceted browsers, we first discuss the root variables that cause such differences and propose that Cognitive Load Theory (CLT) may be able to provide a strong measure of clarity in design, while other existing measures push designers towards richer functionality.

The previous section has indicated that an estimate of CLT should fit nicely into an existing inspection-based evaluation framework, and so our immediate plans are to do so and validate it's findings against user studies of search interfaces. While most of the known methods of reducing CLT can be included in the framework, the *modality* effect may provide the largest challenge, as the framework currently takes no specific note of modality channels. The ultimate test, however, of using CLT this way, will be to actively improve user experiences of exploratory interfaces by providing rich functionality *and* clarity in design.

REFERENCES

1. Back, J. and Oppenheim, C. A model of cognitive load for IR: implications for user relevance feedback interaction. *Information Research 6*, 2 (2001).
2. Capra, R. and Marchionini, G. The relation browser tool for faceted exploratory search. In Proc. *JCDL08*, ACM Press (2008), 420-420.
3. Chandler, P. and Sweller, J. Cognitive Load Theory and the Format of Instruction. *Cognition and Instruction 8*, 4 (1991), 293-332.
4. Chandler, P. and Sweller, J. Cognitive load while learning to use a computer program. *Applied cognitive psychology*, *10*, 2 (1996), 151-170.
5. Hearst, M.A. Next generation web search: setting our sites. *IEEE Data Engineering Bulletin*, *23,* 3 (2000).
6. Hu, P.J.H., Ma, P.C. and Chau, P.Y.K. Evaluation of user interface designs for information retrieval systems: a computer-based experiment. *Decision Support Systems 27*, 1-2 (1999), 125-143.
7. Mu, X. Smartlinks in a Video-Based Collaborative Distance Learning System: a Cognitive Model and Evaluation Study. *School of Information and Library Science*, University of North Carolina, Chapel Hill, 2004.
8. schraefel, m.c., Karam, M. and Zhao, S., Listen to the Music: Audio Preview Cues for the Exploration of Online Music. in *Proc. Interact*, (2003).
9. schraefel, m.c., Wilson, M.L., Russell, A. and Smith, D.A. mSpace: improving information access to multimedia domains with multimodal exploratory search. *Commun. ACM 49*, 4 (2006), 47-49.
10. Schwartz, B. *The Paradox of Choice: Why More Is Less*. Harper Perennial, 2005.
11. Tunkelang, D., Guided Summarization. in *ECIR08 Industry Day Presentation*, (2008).
12. Vicente, K.J. Cognitive engineering: A theoretical framework and three case studies. *International Journal of Industrial and System Engineering 1,* 1 (2006), 168-181.
13. White, R.W., Kules, B., Drucker, S.M. and schraefel, m.c. Introduction. *Commun. ACM 49,* 4 (2006).
14. White, R.W., Marchionini, G. and Muresan, G. Evaluating exploratory search systems Introduction. *Inf. Process. Management 44*, 2 (2008)
15. Wilson, M.L., André, P. and schraefel, m.c., Backward Highlighting: Enhancing Faceted Search. in *Proc. UIST08*, ACM Press (2008).
16. Wilson, M.L., schraefel, m.c. and White, R.W. Evaluating Advanced Search Interfaces using Established Information-Seeking Models. *JASIST*. (to appear).
17. Yee, K.-P., Swearingen, K., Li, K. and Hearst, M., Faceted metadata for image search and browsing. in *Proc. CHI03*, ACM Press (2003), 401-408.
18. Zhang, J. and Marchionini, G., Evaluation and evolution of a browse and search interface: relation browser. in *Proc. National Conference. on Digital Government Research*, (2005), 179-188.

Supporting Exploratory Search for the ACM Digital Library

Vladimir Zelevinsky
vzelevinsky@endeca.com

Joyce Wang
jwang@endeca.com

Daniel Tunkelang
dt@endeca.com

Endeca
Cambridge, MA

ABSTRACT

The Association for Computing Machinery (ACM) is the world's largest educational and scientific computing society, providing the computing field's premier digital library. Many of its articles are tagged by authors with key words and phrases. Unfortunately, the tagging is sparse and inconsistent. As a result, the use of tags for article retrieval leads to high precision but low recall. The alternative of performing full-text search on the tags leads to unacceptably low precision. We have developed a system to bootstrap on author-supplied tags, thus improving tagging across the collection. Preliminary testing suggests we have achieved an order of magnitude increase in recall without perceptibly sacrificing precision. The system can thus leverage the automatically assigned tags to support exploratory search.

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

exploratory search, digital libraries, tagging

1. INTRODUCTION

Digital libraries are increasingly playing a key role in serving the information needs of user communities, particularly communities focused on science and engineering. For example, the Institute of Electrical and Electronics Engineers (IEEE) operates the IEEE Xplore digital library [1] to provide access to its collection of literature in electrical engineering, computer science, and electronics. Similarly, Elsevier operates Scirus [2] in order to provide access to its digital library of scientific research.

The Association for Computing Machinery (ACM) is one of the world's largest educational and scientific computing societies providing the field's premier digital library. This library comprises over a million articles, representing a diverse collection of journals, conference proceedings, and other publications. The Alexa directory lists the ACM web site and its online portal [3] as the two most popular computer science sites on the web [4].

2. ARTICLE TAGGING

2.1 Author Tagging of Articles

The ACM provides the ACM Computing Classification System (CCS) taxonomy that authors can use to generally describe their articles, as well as a set of 16 "General Terms" that apply to all areas of computer science [5]. But the most valuable metadata that authors provide comes in the form of additional key words and phrases that are outside the controlled vocabulary of the CCS and general terms. As guidance for selecting tags, the ACM recommends that authors ask themselves, "Would someone look for this key word or phrase in an index?"

Because this process is uncontrolled, and perhaps because the articles in the ACM digital library are aggregated across a diverse collection of sources, the tagging is sparse and inconsistent. On one hand, only about half of the articles have even a single author-supplied tag. On the other hand, there are over 600,000 distinct tags—the majority of tags are only used once.

Because authors tag their own articles, the tags are consistently accurate as descriptors. In information retrieval terms: if someone were to enter a tag as a search and retrieve those articles assigned the tag by their authors, the results would have high precision.

Unfortunately, those same results would suffer from low recall, not only because about half of the articles are not tagged, but also because tagging introduces what Furnas calls the "vocabulary problem" [6]. Different authors apply different tags to describe the same concept, thus leading to a fragmentation of the vocabulary. Moreover, authors tend to use highly specific tags that make sense in the context of their narrow areas of expertise, but are not necessarily as helpful to less specialized information seekers.

2.2 Tagging and Exploratory Search

Let us enumerate some use cases where we would expect tagging to be helpful:

- Retrieving the articles about a particular topic.
- Identifying the topics related to an article or author.
- Determining which topics express an information need.

Other than perhaps the first of these use cases, the motivation for tagging is largely to support exploratory search. Unfortunately, for the reasons described earlier, the author-supplied tags, despite their accuracy, are not particularly helpful for information seeking in general and for exploratory search in particular.

2.3 Pruning the Author Tags

A key step towards improving the tagging was to reduce the over 600,000 distinct tags to a more manageable vocabulary.

First, we pruned the set by keeping only tags that authors used at least 10 times in the collection, as a first step to leverage the "wisdom of crowds" to identify useful terms. We then normalized the tags to consolidate near-duplicate terms that differed only in case (i.e., uppercase vs. lowercase) or in the inflection of their head word (e.g., operating system, operating systems). We also eliminated tags that were subphrases of other tags (e.g., feature, feature extraction) when the subphrase had lower frequency than the containing phrase—the justification being that a useful subphrase tag should be broader and hence more frequently applicable than the containing phrase. Finally we removed about 100 words manually (e.g., data, algorithm) because we felt their semantic meaning was too broad.

The result of this pruning process was a set of about 10,000 tags.

2.4 Automatic Tagging

We then used a statistical tagging method to apply this pruned set of tags to the collection of articles.

For each article, we identified the tags that occurred in its abstract, normalizing by case and the inflection of the head word as described earlier. We then computed the TF*IDF score of each of the occurring tags and kept those with scores above 90% of the median among the tags. This heuristic reflects our experience that the distribution of TF*IDF scores for terms in a document tend to break into three distinct parts: a head of terms with TF*IDF that are highly topical, a middle region of terms that are somewhat informative, and the tail of terms that are mentioned in passing but are not informative. By using 90% of the median TF*IDF scores as a threshold, we generally capture the informative terms.

3. SYSTEM

We built a prototype in order to empirically test the tagging approach described in the previous section. Because our tagging approach relies on matching tags in the text of article abstracts, we restricted our attention to a subset of about 600,000 articles for which the ACM could provide abstracts.

Our prototype, shown in the screen shot below, highlights the difference between author-supplied tags and tags automatically assigned to articles by our system. We perform text searches against the title, abstract, and tags, but not the full article text.

Shown in the screenshot are the search results for the quoted phrase "exploratory search" (i.e., all articles that match on the exact phrase). There are 128 matching articles, 60% of which have at least one author-supplied tag. We note that our automated tagging assigned at least one tag to 98% of the articles. The snippets shown with each matching article are query-independent summaries that show the context for author-supplied (orange) and automatically assigned tags (blue) occurring in that article. The user can click on a tag to narrow the results to only those including the selected tag.

The pane on the left allows the user to refine the results by facets, such as Author or Publication Year. We have highlighted the two sets of author-supplied tags and automatically assigned tags. In the "split colored tags" view shown, we see the 10 tags from each tag facet with the highest frequency in the current results.

A few observations:

- Even allowing for case variation, only 9 articles are tagged by authors with "exploratory search". While we have no gold standard to tell us how many articles should have been assigned this tag, this number seems extremely low. Note that 9 is below the threshold for inclusion in the vocabulary for automatic assignment.
- Because of the sparsity of the author-supplied tags, only 4 tags occur even 3 times in this set (and two of those are case variants of "exploratory search"). The next 6 author tags look almost random. In contrast, all of the 10 most frequent automatically assigned tags have frequency of at least 9, and are relevant to the results.
- The automatically assigned tags offer useful concepts, such as "search" and "interface" that authors rarely use because they are too general. In the entire collection (not shown), "search" occurs 224 times as an author-supplied tag and 12,887 times as a automatically assigned tag; "interface" occurs 229 times (even allowing for stemming) as an author-supplied tag and 7,051 times as a automatically assigned tag. These broad tags, while rarely supplied by authors, can be very useful as refinements for exploratory search.

4. EVALUATION

The ideal way to evaluate an exploratory search tool would be through a user study, but we have not had the opportunity to conduct such a study. We can, however, consider the quality of the automatic tagging form an information retrieval perspective.

We measure the quality of the automatic tagging as follows: if someone enters a tag as a search and retrieved those articles assigned the tag, the results should correspond to all of the articles and only those articles about the topic represented by the tag. We can hence characterize the performance of our tagging in terms of precision and recall, precision being the fraction of actual tag assignments that are accurate and recall being the fraction of ideal tag assignments that actually occur.

Because we have no gold standard by which to judge the accuracy of our tagging, it is not immediately clear how we can compute precision and recall. All we can rely on as ground truth are the author-supplied tags.

We thus make an assumption that is born out by our experience: that the author-supplied tags have essentially perfect precision. That is, authors almost never assign irrelevant tags to their articles, and hence we consider the precision of author-supplied tags to be 1. In contrast, we make no assumptions about the recall of author-supplied tags, other than that it is low.

We now assert that our automatically assigned tags largely preserve precision while dramatically increasing recall. How do we justify this assertion?

4.1 Precision

Since we do not have assessors to validate our precision claims, we take a data-driven approach that bootstraps on our assumption that the author-supplied tags have essentially perfect precision. We focus on the most frequently assigned author tags, since these allow us to perform meaningful statistical analysis. In all of our analysis, we consolidate near-duplicate tags that differ only in case or in inflection of the head word.

We summarize a set of articles by determining the author-supplied tags most frequently assigned to articles in that set. If two sets of articles are topically similar, we expect high overlap in the sets of frequent author-supplied tags.

In particular, we can compare the set of articles to which authors assigned a particular tag with the set of articles to which we automatically assigned that tag.

The table below shows three statistics comparing the author assignment of tags with their automatic assignment. The "Overlap @ 5" column signifies the number of common tags in the intersection of the five most frequently occurring tags for each article set (i.e., the set of articles to which authors assigned the tag and the set of articles to which we automatically assigned that tag). The "Overlap @ 10" is analogous, only that we consider the ten tags from each set rather than five. Finally, the "Cosine" column computes the angle between the normalized ($|v| = 1$) frequency vectors of the union of the top ten tags for both sets.

Tag	Overlap@5	Overlap@10	Cosine
database	2	6	0.382
xml	4	5	0.991
data mining	5	8	0.997
neural network	3	5	0.979
optimal control	4	7	0.997
electronic commerce	5	6	0.991
computer architecture	2	4	0.983
mobile robot	5	7	0.991
path planning	4	8	0.990
network security	4	6	0.978
parallel algorithms	3	6	0.993
packet switching	3	6	0.941
decision tree	4	4	0.974

While this analysis is crude, it at least provides favorable evidence for our assertion that precision is preserved. We attribute the divergence of the statistics for the "database" tag to the polysemic nature of the term; the other tags are comparatively unambiguous.

4.2 Relative Recall

If we assume that precision is preserved, then it is easy to reason about relative recall: we simply look at the ratio between the number of articles to which we automatically assign a tag and the number of articles to which authors assigned that tag.

For the approximately 10,000 tags in the pruned vocabulary for automatic assignment, this ratio ranges from slightly less than 1 (for less than 1% of the tags) to over 100 for tags like "search" and "computability" that represent broad concepts. The median ratio was 9.0, suggesting an order of magnitude increase in relative recall for the automatically assigned tags, as compared to the author-supplied tags.

5. CONCLUSIONS AND FUTURE WORK

Our goal in working with the ACM Digital Library was to build a practical system that supports exploratory search. While we feel that exploratory search is broadly applicable across many domains, we see it as particularly useful to researchers working with digital libraries.

Our empirical results, while limited by our lack of user studies or assessors for our automatically assigned tags, are very encouraging. Moreover, our own experience with this system is quite positive, and we are working with the ACM to make this tagging available to the broader ACM membership.

Our planned future work is in two areas. First, we would like to evaluate our system more rigorously, both through user studies and through statistical analysis. Second, we would like to apply query expansion and other techniques to further increase recall. We expect that doing so will lead to a precision-recall trade-off, but we feel that we can substantially increase recall without making a comparable sacrifice of precision.

In addition, recall could be further increased by mapping author keywords to a standard set of terminology (for example, ACM Classification terms). Another possibility is either utilizing a database of synonyms or hyponyms, or creating one on the fly. In general, we suggest it would be useful to relate folksonomy terms to those available in controlled vocabularies.

6. ACKNOWLEDGEMENTS

We are grateful to the ACM for their support and cooperation.

7. REFERENCES

[1] IEEE Xplore: http://ieeexplore.ieee.org/

[2] Scirus: http://scirus.com/

[3] ACM Portal: http://portal.acm.org

[4] Alexa listing of Most Popular in Computer Science: http://www.alexa.com/browse?&CategoryID=43037

[5] ACM Computing Classification System (CCS): http://www.acm.org/class/1998/

[6] Furnas, G., Landauer, T, Gomez, L., and Dumais, S. (1987). The vocabulary problem in human-system communication. In *Communications of the Association for Computing Machinery*, 30 (11), Nov 1987, 964-971.