HCIR '07

Workshop on Human-Computer Interaction and Information Retrieval

MIT CSAIL, Cambridge, Massachusetts, USA

October 23, 2007

Workshop Proceedings

For this workshop, researchers and practitioners were invited to present ideas, research results, work in progress, and system demonstrations related to the intersection of Human Computer Interaction (HCI) and Information Retrieval (IR). The intent of the workshop is not archival publication, but rather to provide a forum to build community and to stimulate discussion, new insight and experimentation. This workshop proceedings is provided for convenience.

hcir@csail.mit.edu

http://projects.csail.mit.edu/hcir/

Table of Contents

Partitioning the Web: Shaping Online Consumer Choice

Jolie M. Martin & Michael I. Norton
Harvard University

Imagine you want to spend the weekend with your partner in some city at a nice hotel, with reasonable prices, located near the water. Because you have not been to this city before, you decide to visit one of the many online websites that aggregate information about different hotels, allowing you to view ratings for each hotel's various attributes, narrowing the options until you pick your eventual winner. This research explores the ways in which online vendors structure this search process by categorizing the attributes available for viewing, thereby impacting both the search process and, more importantly, consumers' ultimate purchases. For instance, if you happened to visit a website that displayed the ratings given by prior guests to hotel value, we suggest you might be likely to overweight this criteria and underweight your other key criteria – proximity to the water. On the other hand, if the website showed ratings for location, you would be apt to give this criterion more weight, perhaps changing your decision from a cheap option far from the water to a more expensive option closer to the water. In both cases, though your underlying preference (a reasonably-priced hotel near the water) has not changed, the ways in which information is partitioned changes the search process and your stated preference: your ultimate selection of hotel.

We first survey existing websites to demonstrate the wide variability of rating categories even within the same product category, highlighting the seemingly arbitrary way that attributes are partitioned by web designers for use by consumers. In order to demonstrate the impact of such partitioning on consumer choice, we present results from a series of laboratory studies that demonstrate the impact of partitioning on both explicitly stated attribute weightings and implicit attribute weightings in decisions between options. Across two domains – buying a car and finding a date – Studies 1 and 2 demonstrate that individuals report option attributes to be more or less important depending on the way that those attributes are partitioned. In Studies 3 and 4, we show that participants' preferences for options – selecting a hotel or choosing a date – are impacted by how option attributes are partitioned. Note that in all four studies, the total information available was the same in that participants could see each options' rating on each attribute, but it was the manner in which that information was partitioned that drove decisions.

We conclude by describing the implications of these studies for the design of more effective information interfaces. In particular, online vendors can use knowledge about the impact of partitioning to help consumers make better choices – by partitioning information in ways that reflects consumers' preferences – or make choices that are more closely aligned with the vendors' interest – by partitioning information in ways that drive consumers toward some option that the vendor desires (for example, a hotel that is neither cheap nor near the water!).

Record Relationship Navigation:

Implications for Information Access and Discovery

Christina Anderson, Endeca Technologies

Record relationship navigation extends the faceted navigation model by using relationships between records as a basis for information exploration, rather than creating an asymmetric relationship between records and the facets used to describe them. While record relationship navigation can significantly enhance an end user's discovery experience, it also raises a number of information access questions, including what function a search box should serve, the meaning behind summary analytics (such as record counts), and what exactly constitutes a record.

We will demonstrate record relationship navigation in a prototype ecommerce website whose records include products, reviews, purchase orders, and affinity records (i.e. records specifying the affinities between records). In particular, we will apply record relationship navigation to achieve a transparent social navigation experience, where a user can explicitly navigate the combined space of products and reviews. For example, a user can GPS systems that have reviews written about them by 18-24 year olds, or can see which digital cameras received an average rating of 4 or 5 from females 45 years or older and were reviewed in the past 3-6 months.

Our demonstration will highlight the advantages of record relationship navigation, as well as some of the open issues raised by its use in the information discovery process. These open issues include the challenges of specifying the behavior of a search box in the presence of related records of different types, presenting record counts associated with incremental navigation steps (links) when different steps apply to different record types, and scalability, particularly for features like affinity records that have the potential for quadratic blow-up.

Faceted Browsing, Dynamic Interfaces, and Exploratory Search: Experiences and Challenges

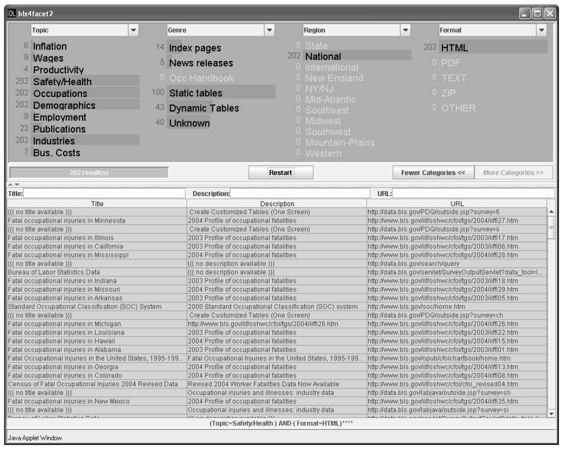
Robert Capra, Gary Marchionini
School of Information and Library Science
University of North Carolina at Chapel Hill
100 Manning Hall
rcapra3@unc.edu, march@ils.unc.edu

Introduction

The Relation Browser (RB) is a graphical interface for exploring information spaces, developed by the Interaction Design Lab at the University of North Carolina at Chapel Hill for use in research on how to support users' needs to understand and explore information. In this abstract, we describe the Relation Browser, results of recent studies, and the design goals for the next-generation RB in current development. At the workshop, we will demonstrate the current RB and a prototype of our next-generation RB.

Current Relation Browser (RB++)

The Relation Browser is designed as a tool for understanding relationships between items in a collection and for exploring an information space (i.e., a set of documents). It has been through a number of major design revisions [2,3]. The current version is called the RB++ (Figure 1). Facets are central to the RB and are displayed at the top of the interface. Results of queries are shown at the bottom of the screen in tabular format. Blue bars and numbers to the left of each facet category indicate how many documents match that category. Previews of queries can be issued by simply mousing over facet categories. For example, mousing over the topic "inflation" will dynamically update the blue bars and number to reflect only documents that are in the inflation topic. By clicking on a facet category and then pressing the "Search" button, results are retrieved and displayed in the lower part of the screen. Results are tightly coupled with the facet categories and with search boxes displayed above each result field. For example, typing "occupations" into the text box above the title field will not only narrow the results shown at the bottom of the screen, but will also update the blue bars and numbers shown at the top of the screen. The current RB allows searching the metadata fields in the result sets, but does not support full-text keyword searches. Because of this, the current RB encourages a "facets first" strategy of exploration.



Title	Description	URL
{{{ no title available }}}	Create Customized Tables (One Screen)	http://data.bls.gov/PDQ/outside.jsp?survey=fi
Fatal occupational injuries in Minnesota	2004 Profile of occupational fatalities	http://www.bls.gov/iif/oshwc/cfoi/tgs/2004/iiffi27.htm
{{{ no title available }}}	Create Customized Tables (One Screen)	http://data.bls.gov/PDQ/outside.jsp?survey=ii
Fatal occupational injuries in Illinois	2003 Profile of occupational fatalities	http://www.bls.gov/iif/oshwc/cfoi/tgs/2003/iiffi17.htm
Fatal occupational injuries in California	2003 Profile of occupational fatalities	http://www.bls.gov/iif/oshwc/cfoi/tgs/2003/iiffi06.htm
Fatal occupational injuries in Mississippi	2004 Profile of occupational fatalities	http://www.bls.gov/iif/oshwc/cfoi/tgs/2004/iiffi28.htm
{{{ no title available }}}	{{{ no description available }}}	http://data.bls.gov/search/query
Bureau of Labor Statistics Data	{{{ no description available }}}	http://data.bls.gov/servlet/SurveyOutputServlet?data_tool=l...
Fatal occupational injuries in Indiana	2003 Profile of occupational fatalities	http://www.bls.gov/iif/oshwc/cfoi/tgs/2003/iiffi18.htm
Fatal occupational injuries in Missouri	2004 Profile of occupational fatalities	http://www.bls.gov/iif/oshwc/cfoi/tgs/2004/iiffi29.htm
Fatal occupational injuries in Arkansas	2003 Profile of occupational fatalities	http://www.bls.gov/iif/oshwc/cfoi/tgs/2003/iiffi05.htm
Standard Occupational Classification (SOC) System	2000 Standard Occupational Classification (SOC) system	http://www.bls.gov/soc/home.htm
{{{ no title available }}}	Create Customized Tables (One Screen)	http://data.bls.gov/PDQ/outside.jsp?survey=ch
Fatal occupational injuries in Michigan	http://www.bls.gov/iif/oshwc/cfoi/tgs/2004/iiffi26.htm	http://www.bls.gov/iif/oshwc/cfoi/tgs/2004/iiffi26.htm
Fatal occupational injuries in Louisiana	2003 Profile of occupational fatalities	http://www.bls.gov/iif/oshwc/cfoi/tgs/2003/iiffi22.htm
Fatal occupational injuries in Hawaii	2004 Profile of occupational fatalities	http://www.bls.gov/iif/oshwc/cfoi/tgs/2004/iiffi15.htm
Fatal occupational injuries in Alabama	2003 Profile of occupational fatalities	http://www.bls.gov/iif/oshwc/cfoi/tgs/2003/iiffi01.htm
Fatal Occupational Injuries in the United States, 1995-199...	Fatal Occupational Injuries in the United States, 1995-199...	http://www.bls.gov/opub/cfoichartbook/home.htm
Fatal occupational injuries in Georgia	2004 Profile of occupational fatalities	http://www.bls.gov/iif/oshwc/cfoi/tgs/2004/iiffi13.htm
Fatal occupational injuries in Colorado	2004 Profile of occupational fatalities	http://www.bls.gov/iif/oshwc/cfoi/tgs/2004/iiffi08.htm
Census of Fatal Occupational Injuries 2004 Revised Data	Revised 2004 Worker Fatalities Data Now Available	http://www.bls.gov/iif/oshwc/cfoi/cfoi_revised04.htm
{{{ no title available }}}	Occupational injuries and illnesses: industry data	http://data.bls.gov/labjava/outside.jsp?survey=sh
Fatal occupational injuries in New Mexico	2004 Profile of occupational fatalities	http://www.bls.gov/iif/oshwc/cfoi/tgs/2004/iiffi35.htm
{{{ no title available }}}	Occupational injuries and illnesses: industry data	http://data.bls.gov/labjava/outside.jsp?survey=si

(Topic=Safety/Health) AND (Format=HTML)****

Figure 1. Relation Browser displaying BLS data

The RB is designed as a generic interface that can accept and display data for many different types of collections. RB instances have been developed for a variety of data sets including U.S. federal statistics (Bureau of Labor Statistics, Energy Information Administration, NSF Science and Engineering Indicators), classical music, the Open Video collection, a university movie database, the CIA World Factbook, and a database of roller coasters. A new version of the RB, called RB07, is currently in development.

Structure and Interaction Study

In the summer of 2006, we conducted two studies to compare three different interface styles (handcrafted web site, simple facet interface, and the Relation Browser) for three different task types (simple lookup, complex lookup, and exploratory search) for the U.S. Bureau of Labor Statistics (BLS) web site data. This data set was fairly large (over 67,000 documents) and semi-structured, providing a good test set for examining facet use for data that does not have a fully defined set of metadata on which to organize.

The BLS web site uses a polyhierarchical structure with two levels of topics displayed on the home page. The design of the BLS site was handcrafted based on a series of needs assessments and user studies. For the simple facet (SF) and Relation Browser (RB) interfaces, we created a facet set using a variety of semi-automated techniques [1]. The results of the studies surprised us: we found no significant differences among the three interfaces for measures of task completion time, accuracy, confidence, or mental effort. The semi-automated facet interfaces (SF & RB) performed just as well as the handcrafted BLS site and no significant two-way interactions between task type and interface were found. These results indicate that facet sets generated using semi-automated methods can provide useful interfaces to large, semi-structured data sets.

Perhaps even more interesting than the quantitative results are the qualitative data and observations we made during the study. One of the common observations was that our participants (recruited from the UNC community, aged 18-35) often attempted to use a "keyword search first" strategy, even in the interfaces that did not directly support this. The BLS web site provided a keyword search feature one click away from the home page, but the SF and RB did not (they both only allowed search on the metadata). Related to this, a number of users expressed feelings of "not being in control" for the RB interface. The current RB strongly emphasizes facets and encourages users to adopt a "facets first" strategy that may be at odds with users' preference for using keyword search first. Despite this conflict, many users appreciated and noted the benefits that facets provide. Thus, we concluded that interfaces should support agile, user-controllable searching and browsing. This has been a design goal for the next-generation of the Relation Browser, described in more detail in the next section.

Next-Generation Relation Browser (RB07)

One of the primary goals of the RB is to provide a tool for exploring data spaces – for gaining a better understanding of the documents and how they are related to each other. Adding full-text search support while maintaining agile exploration is the challenge for our next-generation RB, called RB07.

RB07 is currently in development and we will show a demonstration of the prototype at the workshop. The new design includes flexible facet views so that the user can control how the facets and document counts are presented. The initial two facet views are the traditional view as in the current RB++, and a representation of the facets as a dynamic tag cloud. The new design also includes a choice of ways to view the results of a search. A "grid view" that is similar to the current RB++ results table provides a way to see a concise summary of essential metadata about matching records. A "list view" presents results in a format that is typical of search engines with a document title, matching text snippets, and a URL (if available). The new views for facets and results continue to be tightly coupled using an extensible architecture that can support additional "plug-ins" for displaying facets and results.

Whereas the current RB++ is a pop-up applet that displays in a new window, the new RB07 is designed to be an embedded component of a web page, providing tighter integration with an existing web site if desired. For the implementation of the new RB07, we considered Java and JavaScript as two well-supported client-side languages. JavaScript has easy-to-use access to its surrounding web page, which would have benefits for RB-website integration. However, in the RB, the dynamic updating of the display based on each mouse movement over a facet category triggers a series of computations that are a function of the number of facets, categories, and documents. After extensive testing with JavaScript, we found that it was not fast enough to support the dynamic updating aspects of the RB on current hardware for document collections larger than 5000 to 10000 documents. Thus, we are developing the new RB07 as a Java applet.

Support for full-text keyword searching is being implemented using the Apache Lucene search engine as packaged in the Apache SOLR search server. Although SOLR provides some support for facets, we are not currently leveraging that support, but instead implement facets through the RB itself (this is in large part due to the need to do fast dynamic updates for the mouseovers). Documents are linked between SOLR and the RB07 through a unique document identifier. One of the interface design issues that arose during our development is how to provide clear distinctions between using keywords to search within a result set versus starting a new keyword search. In our current design, new searches are started using a keyword textbox at the top of the display and refinement searches are issued through a textbox closer to the result set. We hope that the new RB07 will help address users expectations of how to explore document spaces, while still providing powerful interface components for seeing relationships in the collection and refining result sets.

Future Questions

While facets are widely regarded as being useful to information seeking, especially in large data sets with complete metadata that can be used as facets (such as shopping domains), a number of research questions still remain: How are facets used during the information seeking process? When, how, and why are facets helpful (i.e. facets first versus facets to refine)? How is facet use affected by task type? What role do facets play in exploring and gaining an understanding of the information space? Do facets help users refind documents they have seen before, acting as waypoints?

Understanding these issues will help us create better tools for information discovery and exploratory searching and we expect that user studies and other empirical investigations will lead to answers to these questions and guide future system designs.

References

[1] Capra, R., Marchionini, G., Oh, J. S., Stutzman, F., and Zhang, Y. (2007). Effects of structure and interaction style on distinct search tasks. In Proceedings of the 2007 Conference on Digital Libraries (Vancouver, BC, Canada, June 18 - 23, 2007). JCDL '07.

[2] Marchionini, G. & Brunk, B. (2003). Toward a General Relation Browser: A GUI for Information Architects. Journal of Digital Information, 4(1), http://jodi.ecs.soton.ac.uk/Articles/v04/i01/Marchionini/

[3] Zhang, J., and Marchionini, G. (2004). Coupling Browse and Search in Highly Interactive User Interfaces: A Study of the Relation Browser++. Proceedings of the 4th ACM/IEEE Joint Conference on Digital Libraries (Tucson, AZ: June 7-11, 2004), 384.

Idea Navigation: Structured Browsing for Unstructured Text

Robin Stewart
MIT
stewart@csail.mit.edu

Gregory Scott
Tufts University
Greg.Scott@tufts.edu

Vladimir Zelevinsky
Endeca
vzelevinsky@Endeca.com

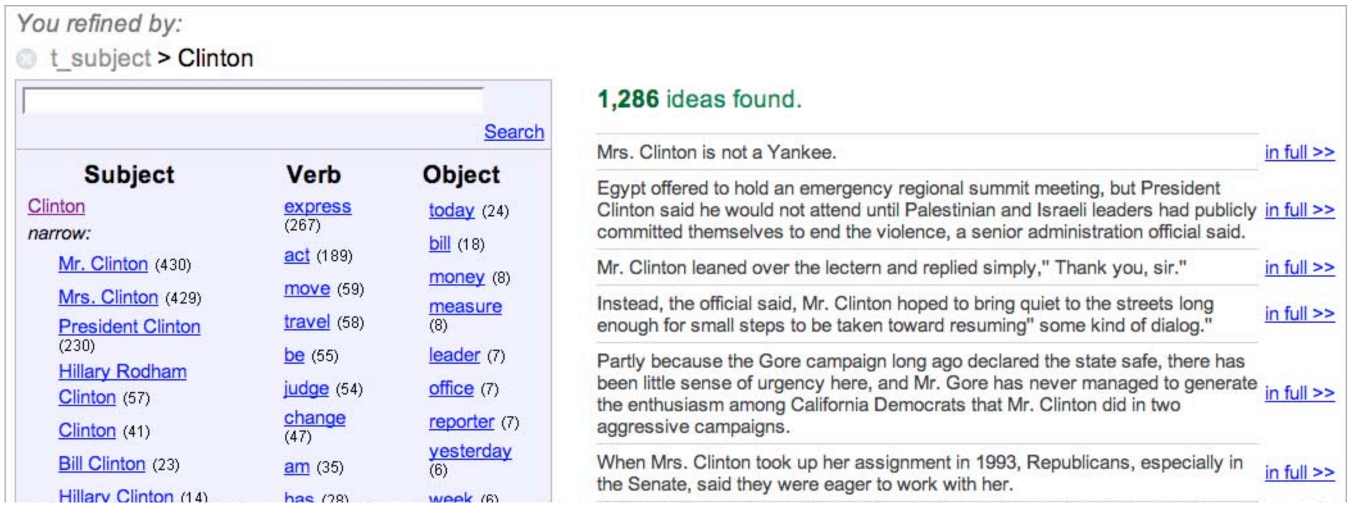
INTRODUCTION

It is well established that the omnipresent search box is insufficient for supporting many common information-seeking tasks and strategies [1]. In order to provide a better interface, many commercial websites now use faceted browsing, which provides the ability to narrow a search result set by choosing to view only a particular slice of metadata [2]. For example, one can search for "televisions" and then narrow the results by clicking on facets such as "flat screen" or "$1500-$3000". This type of interface is particularly useful for exploratory search tasks where users may not know how to define a priori the best query to solve their task – whether because they don't know in advance what information is available in a particular collection or they cannot anticipate which keywords would best describe their desired results.

A number of interfaces have recently been introduced that aim to further expand support for exploratory search over various domains [3]. However, the facets exposed by these interfaces have been limited to explicitly assigned metadata and keywords extracted from text. This limitation is acceptable if the search happens to be a conjunction of the available keywords or metadata, as in a search for a flat-screen television that costs $1500-$3000. But what if we are looking for something more abstract or subjective? For example, we may want to find "historical events that are interesting in the present context" or "quotations that one might find controversial." The available metadata is unlikely to be useful for these tasks.

Further, even when relevant facets do exist, the query may depend on a relationship *between* facets. Consider a search for campaign proposals made by Hillary Clinton. It would not be sufficient to look for documents that contain the terms "Hillary Clinton" and "proposals", because the result set will contain many documents in which Clinton is not the person doing the proposing. This type of query is particularly common in scientific, legal, and patent searches (e.g.: find molecules that target a particular cell; find inventions that burn solids for locomotion). Current interfaces make such search tasks awkward at best, since users need to scan through the list of matching articles for passages that might indicate relevance.

To help users better answer these types of queries, we have developed a richer summarization of natural language text documents that takes into account the semantic information provided by English sentence structure. Our system initially scans every sentence in the document collection to extract sets of terms (subject–verb–object) that indicate the



You refined by:
t_subject > Clinton

Search

Subject	Verb	Object
Clinton	express (267)	today (24)
narrow:	act (189)	bill (18)
Mr. Clinton (430)	move (59)	money (8)
Mrs. Clinton (429)	travel (58)	measure (8)
President Clinton (230)	be (55)	leader (7)
Hillary Rodham Clinton (57)	judge (54)	office (7)
Clinton (41)	change (47)	reporter (7)
Bill Clinton (23)	am (35)	yesterday (6)
Hillary Clinton (14)	has (28)	week (6)

1,286 ideas found.

Mrs. Clinton is not a Yankee. in full >>

Egypt offered to hold an emergency regional summit meeting, but President Clinton said he would not attend until Palestinian and Israeli leaders had publicly committed themselves to end the violence, a senior administration official said. in full >>

Mr. Clinton leaned over the lectern and replied simply," Thank you, sir." in full >>

Instead, the official said, Mr. Clinton hoped to bring quiet to the streets long enough for small steps to be taken toward resuming" some kind of dialog." in full >>

Partly because the Gore campaign long ago declared the state safe, there has been little sense of urgency here, and Mr. Gore has never managed to generate the enthusiasm among California Democrats that Mr. Clinton did in two aggressive campaigns. in full >>

When Mrs. Clinton took up her assignment in 1993, Republicans, especially in the Senate, said they were eager to work with her. in full >>

Figure 1. The full system interface after clicking on "Clinton". Further refinement options are on the left. Sentences from the document collection that have a Clinton as their subject appear on the right.

presence of what we term an **idea**, such as "Clinton–proposed–reforms." It then groups similar terms together under broader categories so that they can be more effectively summarized. The system then dynamically aggregates all of the ideas in response to user input, allowing navigation through the corpus using the same interaction style as faceted browsing (figure 1). This gives users (a) a view into the most common ideas in the result set (since they are presented with the list of concepts extracted from this set) and (b) an easy way to narrow in on concepts that look interesting. We call this process **idea navigation**.

IDEA NAVIGATION

We demonstrate idea navigation by answering one of the questions above: What did Hillary Clinton propose in previous campaigns? Our prototype system contains all ~9000 articles published in October 2000 by a popular news source. At that time, Clinton was running for the Senate. We first select "Clinton" in the Subject column, revealing a variety of narrow terms such as "Mr. Clinton" and "President Clinton" (figure 1). We choose "Mrs. Clinton" to refine the results to sentences containing Mrs. Clinton as their subject. Selecting the broad action "express" in the Verb column refines the result set to things that Mrs. Clinton said. This results in 117 matching ideas, so we select the narrow verb "propose". We have now narrowed our result set down to five sentences, all of which provide answers to our query. One is: "Mrs. Clinton, for example, has proposed federally financed scholarships for college students who commit to teaching."

We chose to represent ideas as sets of three terms (subject–verb–object) because this representation has been successfully used in question answering systems to help answer focused questions such as "Who did Hillary Clinton marry?" [4]. This representation is also the basic underpinning of the semantic web's Resource Description Framework. Numerous interfaces for searching such semi-structured data have been built, including ESTER [5], which proactively displays refinement options that it considers relevant. However, to our knowledge, our system is the first to present the ternary representation as a faceted browsing interface.

EVALUATION

To better understand the extent to which Idea Navigator is understandable and helpful for information seeking, we carried out a formative evaluation with 11 participants. Presumably due to the dominance of keyword search interfaces today, most subjects had a clear initial bias towards formulating queries as keywords in the provided search box, such as "roger clemens throw bat" or "offensive statement." Even so, subjects consistently and successfully used the idea navigation refinements to improve upon their search box results. As the session progressed, they tended to use these refinements more and more. Across all subjects and all tasks, a total of 100 idea navigation refinements were performed, versus only 61 searches. This is strong evidence that users understood the new interface, expected it to be useful, and continued to use it.

CONCLUSIONS AND FUTURE WORK

In this paper, we demonstrate how document search interfaces could be enhanced by extending faceted browsing to the subject–verb–object representation of ideas. Our user study demonstrated that such an interface is understandable to first-time users and useful for solving realistic search tasks that are poorly supported by existing systems.

Future work includes: increasing the number of sentence types that our system understands; trying different ways of grouping the idea components; improving the interface design; and exploring alternate idea representation schemes that better handle adjectives, prepositional phrases, or other natural language information. We plan to integrate idea navigation into a full-featured search interface with a large health science document set, allowing us to perform a more extensive, comparative user study that examines user performance when various search components are available.

Beyond this, we would like to further investigate the use of idea navigation as an interface for exploring facts, either automatically extracted from documents or entered manually as semantic web data. Just as faceted browsing supports exploratory search in document collections, we posit that the techniques of idea navigation may well support the related task of exploratory question answering.

REFERENCES

1. White, R. W., Kules, B., Drucker, S. M., schraefel, m. c. (2006) Supporting Exploratory Search, Introduction, Special Issue. In *Communications of the ACM* 49(4), pp. 36-39.
2. Hearst, M., English, J., Sinha, R., Swearingen, K., and Yee, K.-P. (2002) Finding the Flow in Web Site Search. In *Communications of the ACM* 45(9), pp. 42-49.
3. Wilson, M., schraefel, m.c., White, R. (2007) Evaluating advanced interfaces using established information-seeking models. Technical Report, School of Electronics and Computer Science, University of Southampton. http://eprints.ecs.soton.ac.uk/13737/
4. Katz, B., Lin, J. (2003) Selectively Using Relations to Improve Precision in Question Answering. In *Proc. EACL 2003 Workshop on Natural Language Processing for Question Answering*.
5. Bast, H., Chitea, A., Suchanek, F., Weber, I. (2007) ESTER: Efficient Search on Text, Entities, and Relations. In *Proc. SIGIR 2007*, ACM Press, pp. 671-678.

GK: A post-search information retrieval system

Joseph Barillari[1]

This abstract introduces GK, a web-based software system for research support. GK is designed to help users store, organize, and navigate large quantities of text, currently targeting but by no means limited to biomedical research.[2]

Motivation. Information retrieval researchers have addressed with great success the search problem: very roughly speaking, the process of retrieving a small collection of relevant documents from a collection many orders of magnitude larger. GK addresses a different part of information retrieval: the *post-search* problem. After he or she collects several hundred to a few tens of thousands of documents that a search engine has indicated were relevant, GK helps the user make sense of them.

Use case. GK's canonical use case involves a biomedical researcher (we'll call her U.) investigating a new field. PubMed, Google Scholar, and ISI Web of Science[3] will happily locate a few hundred relevant papers, but what does U. do with them? She'd like to find the most interesting parts of the papers without reading everything (since there are bound to be redundancies and irrelevant pieces). U. begins by dragging the papers into her GK volume. (GK has a network filesystem interface, so uploading data is simple.) GK indexes the papers and draws a map of the concepts[4] mentioned therein. The map lets U. navigate the documents horizontally: when she encounters the gene *fgfr2* in paper A., the map shows her that paper B. associates it with *VEGF* and paper C. associates it with *wnt*. The map also shows her the sentences from which it inferred those associations and lets her jump from A. directly to those parts of documents B. and C.

As she reads, U. can highlight and annotate the documents in her collection. She can highlight HTML documents directly within GK, or use Adobe Acrobat to add annotations to PDF files, which GK can read and index. She can then ask GK to draw an editable outline that collects her highlights and annotations from all of the documents in one place.

Design goals. GK is designed to solve the three biggest post-search problems: storage, organization, and navigation.

Storage. Storage is mundane but too-often ignored. An ideal system would let a researcher access his or her collection of documents from any device at any location. GK pro-

[1]Harvard School of Engineering and Applied Sciences and the Harvard-MIT Division of Health Science and Technology. (barillar@fas.harvard.edu)

[2]For simplicity's sake, the examples below are all biomedical.

[3]Scientific search engines. PubMed is run by the National Institutes of Health and indexes the biomedical literature. Google Scholar indexes scientific papers. ISI Web of Science is a citation index: it lets the user search for papers that cite a particular paper.

[4]"Concepts" is defined loosely as "interesting n-grams." Currently, GK identifies interesting concepts using lists of known-interesting terms (for instance, gene names). In other fields, it might use proper nouns (by examining capitalization patterns) or unlikely phrases (c.f. Amazon.com).

vides both a through-the-web and network-filesystem-based access to a user's collection. The filesystem uses the HTTP-based WebDAV protocol, which is supported on all modern platforms. It also supports transactions, versioning, and arbitrary metadata. The web interface provides access to a search system that indexes both the full text and the metadata of the user's documents. (For instance, if a document is indexed in MEDLINE and GK can find its DOI,[5] GK will automatically download its metadata from PubMed and index it.)

Organization. In the physical world, users highlight and annotate papers with pens and file them into folders. GK supports highlighting, annotation, and filing. It also incorporates organizational techniques impossible in the physical world: for instance, the ability to place items in multiple folders (tagging), to search for documents by the notes one has taken on them, and to pull highlights and annotations from a group of documents into an editable outline.[6]

Navigation. Navigation is by far the most difficult and most important post-search problem—given a set of documents, all of which are relevant, how does one help the user locate the sections he or she would find most interesting? GK's key navigational feature is the automated drawing of concept maps which show how how individual fragments of documents relate to one another. This granularity lets users jump directly from a paragraph of interest in one document directly to a paragraph of interest in another. The functions which detect concepts of interest and infer interrelations between them are under heavy development.

Direction. GK is under active development, with a particular emphasis on improving its navigational features. While many IR systems have incorporated graph-based navigational tools, few such tools have gained mass appeal. A key aim for GK at present is to determine if graphs and maps are too cumbersome to be helpful, or if a properly-designed mapping tool can be genuinely useful.

[5]MEDLINE: the NIH's index of the medical literature. PubMed: the web interface for MEDLINE. DOI: Document Object Identifier, a numbering system used by many publishers.

[6]Support for sharing annotations between users is under development.

A Knowledge-Based Search Engine Powered by Wikipedia

David Milne   Ian H. Witten   David M. Nichols

Department of Computer Science, University of Waikato
Private Bag 3105, Hamilton, New Zealand
+64 7 838 4021

{dnk2, ihw, dmn}@cs.waikato.ac.nz

Introduction

This paper describes Koru: a new search interface that offers effective domain-independent knowledge-based information retrieval [1]. Koru exhibits an understanding of the topics involved in both queries and documents, allowing them to be matched more accurately. It helps users express queries more precisely and evolve them interactively. This understanding is mined from the vast investment of manual effort and judgment that is Wikipedia. This open, constantly evolving encyclopedia yields manually-defined yet inexpensive structures that can be specifically tailored to expose the topics, terminology and semantics of individual document collections. This paper describes a brief overview of Koru and the knowledge base it extracts. A more detailed description and evaluation of the system can be found elsewhere [2].

Koru

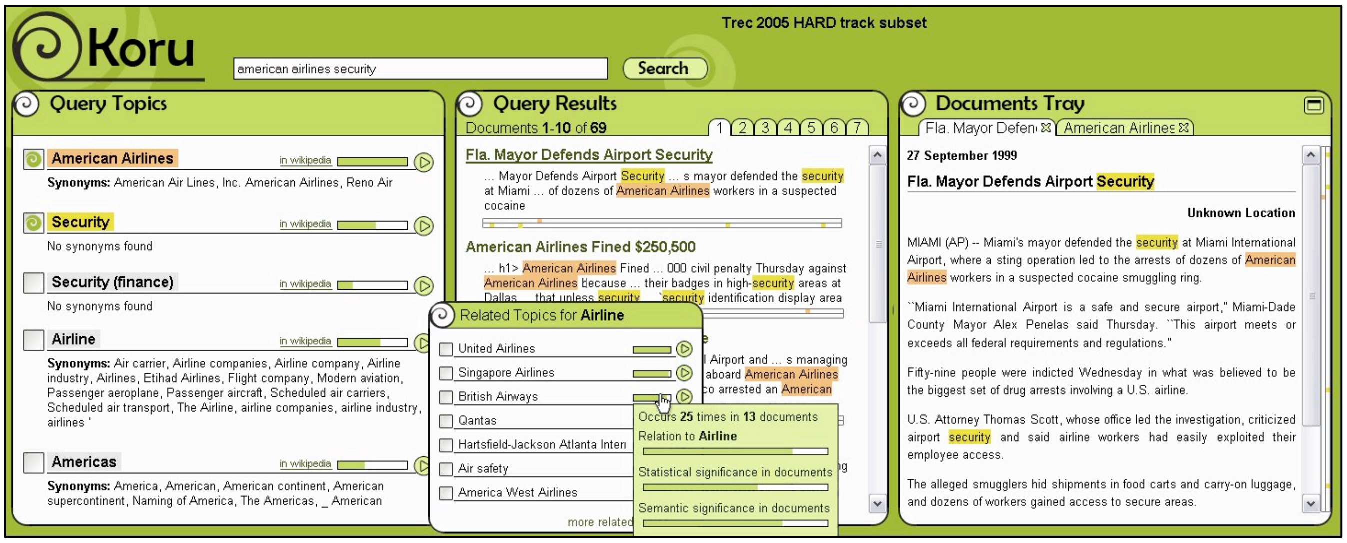
Koru is the Māori word for the newborn, unfurling fern frond; a delicate spiral of expanding fractal shapes. For indigenous New Zealanders it symbolizes growth; expansion; evolution. Likewise, the Koru topic browsing system aims to provide an environment in which users can progressively work towards what they seek.

Koru's interface is illustrated in Figure 1. The uppermost area is a classic search box in which the user has entered the query *american airlines security*. Below are three panels: query topics, query results, and the document tray.

The latter two panels are fairly standard. The query results list documents in much the same fashion as other search engines, while the document tray allows the reader to collect multiple documents they wish to peruse. The first panel—query topics—is intended to display Koru's interpretation of the query and provide a base from which to evolve it.

The query topics listed here—*American Airlines*, *Security*, *Security (finance)*, *Airline* and *Americas*—are identified automatically by checking words and consecutive sequences of words in the query against articles in Wikipedia. Synonyms mined from the same resource are listed below that term. For example, amongst the topic *Airline*'s synonyms are *air carrier*, *airline company,* and *scheduled air transport*. These are (a) used internally to improve queries, and (b) presented to the user to help them understand the sense of the topic. The user can also learn more about a topic by clicking the adjacent Wikipedia link.

Query terms are often ambiguous and relate to multiple entries in Wikipedia. By *security*, for example, the user could also mean property pledged as collateral for a loan. Each sense is included, and ranked according to the likelihood that it is a relevant, significant topic for this particular document collection. Only the



american airlines security

American Airlines
Synonyms: American Air Lines, Inc. American Airlines, Reno Air

Security
No synonyms found

Security (finance)
No synonyms found

Airline
Synonyms: Air carrier, Airline companies, Airline company, Airline industry, Airlines, Etihad Airlines, Flight company, Modern aviation, Passenger aeroplane, Passenger aircraft, Scheduled air carriers, Scheduled air transport, The Airline, airline companies, airline industry, airlines '

Americas
Synonyms: America, American, American continent, American supercontinent, Naming of America, The Americas, _ American

Documents 1-10 of 69

Fla. Mayor Defends Airport Security
... Mayor Defends Airport Security ... s mayor defended the security at Miami ... of dozens of American Airlines workers in a suspected cocaine

American Airlines Fined $250,500
... h1> American Airlines Fined ... 000 civil penalty Thursday against American Airlines because ... their badges in high-security areas at Dallas ... that unless security ... `security identification display area

l Airport and ... s managing aboard American Airlines co arrested an American

Related Topics for Airline
United Airlines
Singapore Airlines
British Airways
Qantas
Hartsfield-Jackson Atlanta Intern
Air safety
America West Airlines

Occurs 25 times in 13 documents
Relation to Airline
Statistical significance in documents
Semantic significance in documents

27 September 1999
Fla. Mayor Defends Airport Security
Unknown Location

MIAMI (AP) -- Miami's mayor defended the security at Miami International Airport, where a sting operation led to the arrests of dozens of American Airlines workers in a suspected cocaine smuggling ring.

``Miami International Airport is a safe and secure airport," Miami-Dade County Mayor Alex Penelas said Thursday. ``This airport meets or exceeds all federal requirements and regulations."

Fifty-nine people were indicted Wednesday in what was believed to be the biggest set of drug arrests involving a U.S. airline.

U.S. Attorney Thomas Scott, whose office led the investigation, criticized airport security and said airline workers had easily exploited their employee access.

The alleged smugglers hid shipments in food carts and carry-on luggage, and dozens of workers gained access to secure areas.

Figure 1: Browsing Koru for topics and documents related to *american airlines security*

	Baseline	**Koru**
Recall	43.4%	51.5%
Precision	10.2%	11.6%
F-measure	13.2%	17.3%

Table 1: performance of tasks

top-ranked topics that cover all the query terms are used for retrieval (in the example, *American Airlines* and the first meaning of *security*). This can be overridden manually using the checkboxes to the left of each topic.

Topics which are recognized in the query can be investigated in isolation by using them to browse Wikipedia for other topics of interest. In Figure 1 the user has chosen to expand topics related to *airline* (by clicking the triangle to the right), and can investigate further topics of interest such as *Singapore Airlines* and *British Airways*. Any of these could be incorporated into the query by clicking the appropriate checkbox. As with alternate senses, these topics are ranked according to their expected relevance.

What the figure does not convey is that to avoid clutter not all the panels in Figure 1 are visible at any given time. Initially only the first two are shown. The user builds an effective query by adding and removing phrases and topics until the query and resulting list of documents satisfies the user's information need. Once a suitable query is formed, the user must determine the most relevant ones and judge whether they warrant further reading. At this point the panels slide across so that only the query results and document tray are visible.

Creating a Relevant Knowledge Base

To work well, Koru relies on a large and comprehensive knowledge base. From Wikipedia we derive a thesaurus that is specific to each particular document collection. Wikipedia is particularly attractive for this work because it represents a vast domain-independent pool of manually defined terms, concepts and relations. By intersecting this with individual document collections, we are able to provide thesauri that are individually tailored to those who seek knowledge from the documents.

The basic idea is to use Wikipedia's articles as the building blocks of a knowledge base and its skeleton structure of hyperlinks to determine which blocks we need and how these should fit together. Each article describes a single concept; its title is a succinct, well-formed phrase that resembles a term in a conventional thesaurus—and we treat it as such. Concepts are often referred to by multiple terms— e.g. in Figure 1 *airline* is grouped with *air carrier,* and *passenger aircraft*—and Wikipedia handles these using "redirects": pseudo-articles that exist only to connect an alternative title of an article with the preferred one. Related topics—*british airways, qantas* and *air safety*—are mined from the hyperlinks within Wikipedia's article on airlines, and the categories in which it is placed.

The danger in using Wikipedia's structure is that because it is so huge (~2 million topics, plus a further 2 million synonyms) the Koru user will become swamped with irrelevant topics and links. It is essential to identify the concepts relevant a particular document collection, and place these in a structure that allows navigation between related concepts. This process is described in detail in [2].

Evaluation

To gain insight into the performance of Koru for document retrieval, we conducted an experiment in which participants performed tasks for which the relevant documents had been identified manually. These tasks were specifically selected to encourage a high degree of interaction. To provide a baseline we created another version that provides as much of the same functionality as possible without using a thesaurus, and whose interface is otherwise identical. Comparison of these two systems provided concrete evidence of the effectiveness of Koru and Wikipedia for assisting information retrieval.

Due to Wikipedia's use of contemporary language and exceptional size, Koru was able to recognize and expand upon almost all queries that were issued. This assistance was effective; as shown in Table 1, it resulted in significant improvements in F-measure.

Koru's design was also validated, in that it allowed users to apply the knowledge found in Wikipedia to their retrieval process easily, effectively and efficiently. The following quote, given by one participant at the conclusion of their session, summarizes Koru's performance best:

> *It feels like a more powerful searching method, and allows you to search for topics that you may not have thought of ...*
>
> *... it could use some improvements but the ability to graphically turn topics on/off is useful, and the way the system compresses synonymous terms together saves the user from having to search for the variations themselves. The ability to see a list of related terms also makes it easier to refine a search, where as with keyword searching you have to think up related terms yourself.*

Unfortunately we found that the interactive browsing facilities offered by Koru are significantly flawed. More work is required to identify related topics that are of interest given the context of the user's task at hand, and to allow users to explore them efficiently. Koru currently provides topic recognition and automatic expansion that helps users express their information needs more easily and consistently, but only as a one-step process of improvement—which can only take queries so far. Our goal in future is to improve Koru's interactive query expansion and exploratory searching facilities until it provides this ability to unfurl queries, and thus lives up to its name.

References

[1] http://www.nzdl.org/koru

[2] Milne, D., Witten, I.H. and Nichols, D.M. (2007). A Knowledge-Based Search Engine Powered by Wikipedia. *In Proc. of CIKM'07*, Lisbon, Portugal.

Visual Text Analysis by Lay Users: The Case of “Many Eyes”

Fernanda B. Viégas
IBM Research
viegasf@us.ibm.com

Martin Wattenberg
IBM Research
mwatten@us.ibm.com

Abstract

Many Eyes (http://www.many-eyes.com) is a public web site that provides "visualization as a service," allowing users to upload data sets, visualize them, and comment on each other's visualizations. Among the visualization techniques offered by the site are two aimed at unstructured text: a “tag cloud” view and a “word tree,” a type of visual concordance view. Both techniques have seen heavy usage. Our talk will break into two parts. First, we will introduce and demonstrate the two text visualization techniques. Second, we will describe the patterns of usage we have observed, particularly around political speeches and testimony and artistic expression.

Visualizations

The first text visualization introduced on Many Eyes was a tag cloud, that is, a simple display in which word frequencies in a text are indicated by font size. Tag clouds have become a familiar presence on many web sites. The Many Eyes tag cloud, however, includes distinctive features such as instant letter-by-letter search and a two-word-phrase view. Figure 1 (next page) is an example of a tag cloud on Many Eyes, showing the recent controversial speech by Iran’s president at Columbia University.

Our second text visualization, dubbed a “word tree,” is a new kind of visual concordance. When a user types in a word or phrase, the visualization displays a tree structure showing all the different contexts in which the word has been used. (From a computer scientist’s perspective, it shows a subtree of the suffix tree of the sequence of words in the text.) Figure 2 shows an example, applied to Martin Luther King’s famous “I have a dream” speech. Users can navigate the word tree by typing or by clicking on branches of the tree to zoom, and can see a view of either leading or trailing context for a word.

Usage

What sorts of analyses have users been creating with these visualizations? Probably the most common application is to political texts. The words of George Bush, Gordon Brown, and Nicholas Sarkozy have all come under scrutiny, as has testimony from Alberto Gonzales and Bill Clinton. In many cases, the visualizations are used by amateur pundits to make political points on blogs. We have also seen professionals who use these tools to bolster their arguments, ranging from a well-known journalism professor to employees of a respected foundation that aims for political transparency.

A second common use case involves personal artistic expression. Several users have created miniature art projects based on tag clouds. One user made a sort of visual poem based on the contents of their freezer. Another has created a series of “litmashes” in which he concatenates two novels and visualizes the results. A third began with a political objective—analyzing grants to artists—and ended up with a commission to create an art project based on his tag cloud.

We’ll conclude our talk with a discussion of the implications of these examples. The usage we have seen on our site, as well as the activity on hundreds of blogs that refer to our site, indicates an intense amateur interest in text analysis. We believe this interest points to new directions for visualization, and also suggests there may be unexplored ways in which other types of textual data mining and information retrieval may be used by citizen activists and artists.

Figure 1.

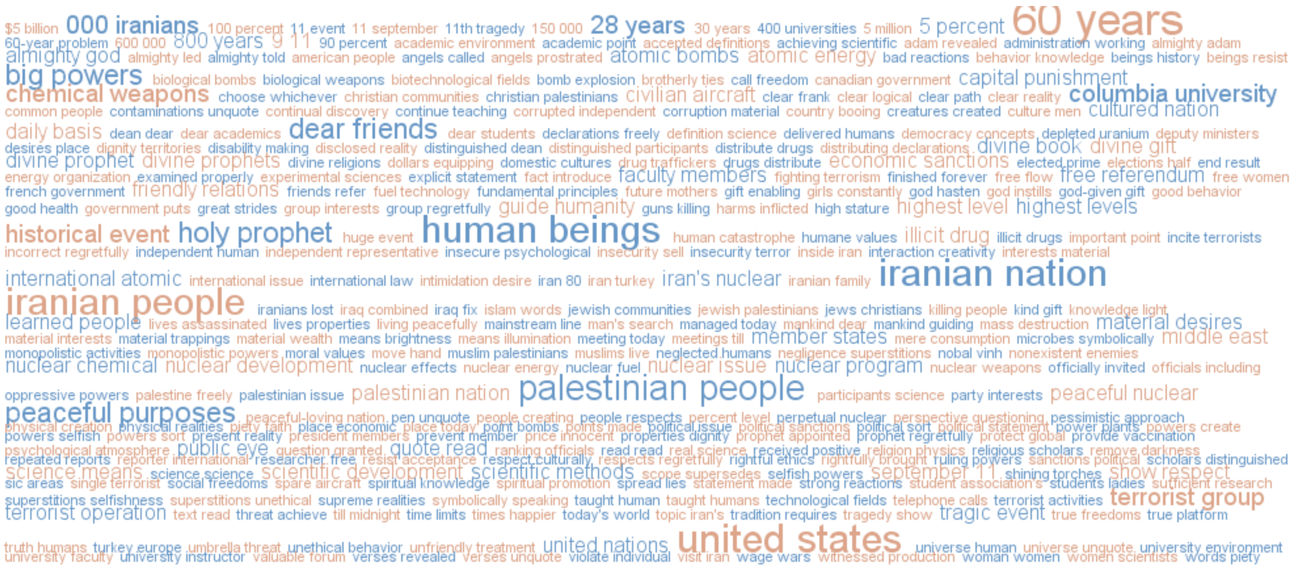
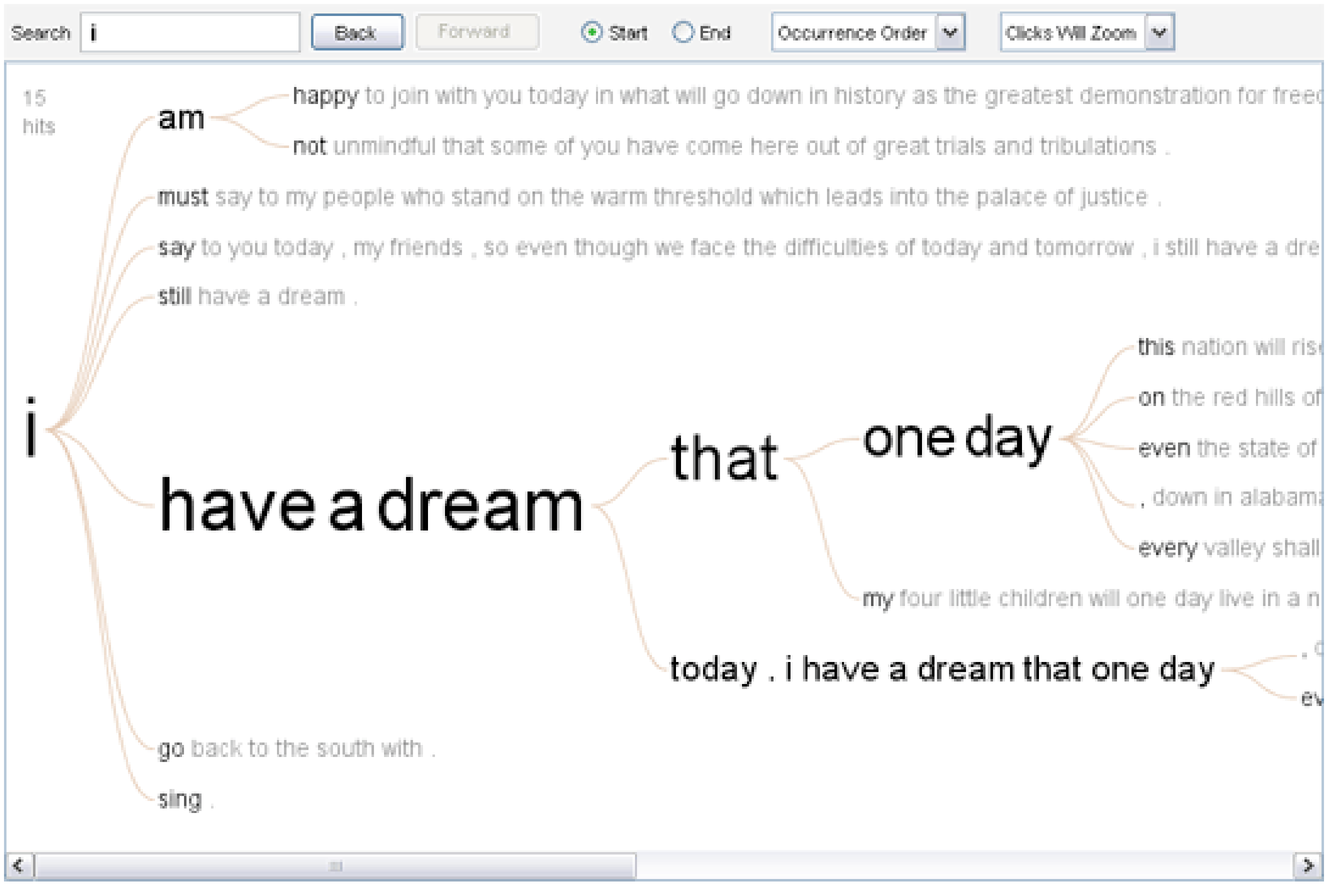


Figure 2.



Search: i

[✓ Start] End

15 hits

i

am happy to join with you today in what will go down in history as the greatest demonstration for freed

am not unmindful that some of you have come here out of great trials and tribulations .

must say to my people who stand on the warm threshold which leads into the palace of justice .

say to you today , my friends , so even though we face the difficulties of today and tomorrow , i still have a dre

still have a dream .

have a dream that one day this nation will ris

have a dream that one day on the red hills of

have a dream that one day even the state of

have a dream that one day , down in alabam

have a dream that one day every valley shall

have a dream that my four little children will one day live in a n

have a dream today . i have a dream that one day , c

have a dream today . i have a dream that one day ev

go back to the south with .

sing .

Personal Information Management, Personal Information Retrieval?

Michael Bernstein, Max Van Kleek, David R. Karger
MIT CSAIL
32 Vassar Street
Cambridge, MA 02139
msbernst@mit.edu, emax@csail.mit.edu
karger@mit.edu

mc schraefel
Electronics and Computer Science
University of Southampton
Southampton, UK
mc+hcir@ecs.soton.co.uk

ABSTRACT

Traditional information retrieval has focused on the task of finding information or documents in a largely unknown space such as the Web or a library collection. In this paper we propose that the space of Personal Information Management (PIM) holds a great number of problems and untapped potential for research at the intersection of HCI and IR. In this position paper we focus on the problem of *information scraps*, or unstructured notes and thoughts, as a particularly interesting space for future research in HCI and IR.

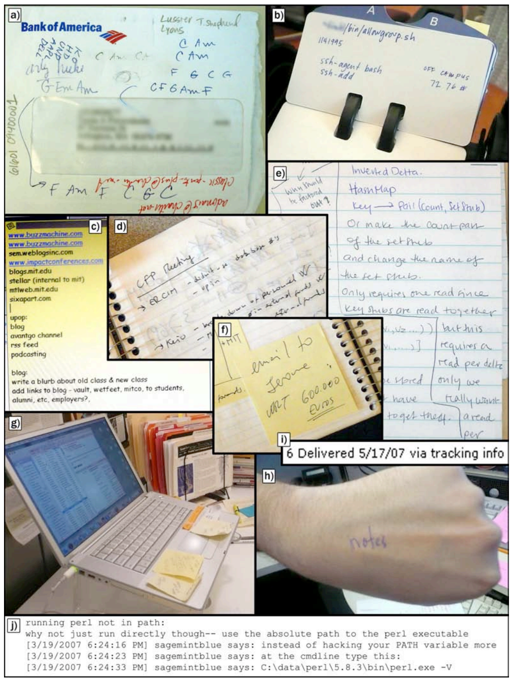
INTRODUCTION

Information retrieval has traditionally assumed a user goal of finding information or documents within a search space whose contents may not be known *a priori* to the user, such as the Web. The user's challenge is to specify an accurate information query (e.g., "What is the capital of Uruguay?") and the system's challenge is to return the most relevant answers or documents.

Contrast this situation with Personal Information Management (PIM). Here, the user's challenge is instead to organize, find and manipulate *his or her own* information, of which the user has intimate knowledge. Though users are still performing information retrieval or search tasks, the tasks' nature may change dramatically. In PIM, users may bring much more highly contextual queries ("When is that meeting with Kerry that I set up while I was at lunch on Friday?"), and have personally authored much of the information they are attempting to retrieve. Further, the user's task does not finish with the retrieval of the document or datum; it often continues through cycles of editing and reorganization.

PIM embeds many IR tasks that users deal with on a daily basis, yet these tasks have been largely overlooked by the IR community. We believe that PIM as an area of research has much to gain from information retrieval techniques and that IR, in turn, may benefit from focusing some of its effort on PIM tasks. In this paper, we outline our research on *information scraps* [1], detailing how this PIM problem interacts closely with information retrieval, and how taking a traditional IR approach might overly decontextualize the problem.

INFORMATION SCRAPS



www.buzzmachine.com
www.buzzmachine.com
sem.weblogsinc.com
www.impactconferences.com
blogs.mit.edu
stellar (internal to mit)
mtlweb.mit.edu
sixapart.com

upop:
blog
avantgo channel
rss feed
podcasting

blog:
write a blurb about old class & new class
add links to blog - vault, wetfeet, mitco, to students, alumni, etc, employers?,

6 Delivered 5/17/07 via tracking info

```
running perl not in path:
why not just run directly though-- use the absolute path to the perl executable
[3/19/2007 6:24:16 PM] sagemintblue says: instead of hacking your PATH variable more
[3/19/2007 6:24:23 PM] sagemintblue says: at the cmdline type this:
[3/19/2007 6:24:33 PM] sagemintblue says: C:\data\perl\5.8.3\bin\perl.exe -V
```

Figure 1. Information Scraps collected in our investigation of existing practice.

Despite the number of personal information management tools available today, a striking amount of our data remains out of their reach: the content is instead scribbled on Post-it notes, scrawled on corners of sheets of paper, buried inside the bodies of e-mail messages sent to ourselves, and typed haphazardly into text files (Figure 1). This scattered data contains our great ideas, sketches, notes, reminders, driving directions, and even our poetry. We refer to these pieces of personal information as *information scraps*.

We conducted a study consisting of 27 semi-structured interviews and artifact examinations of participants' physical and digital information scraps, at 5 different organizations [1]. Here, we summarize one piece of the study in support of our argument, focusing on general uses our participants found for information scraps.

Roles that Information Scraps Play

We consolidated a list of common information scrap roles from participants' responses regarding how and why they chose to store information of various types in scraps.

Temporary Storage. Information scraps' small, discardable presence enabled their common use as temporary storage. One participant kept Post-it notes on her laptop palm rest for just this purpose, recording visitors' names and contact information, later to be disposed of.

Archiving. Many information scraps were intended to reliably hold on to important personal information for long periods of time. Participants commonly used information scraps to archive notes from meetings and passwords.

Work-in-progress. Our participants shared with us many work-in-progress scraps, such as half-written emails, ideas for business plans, brainstorms, and interface designs. "Before I put anything in the computer, I like to put it on the whiteboard first," one participant explained of her newsletter layout design process.

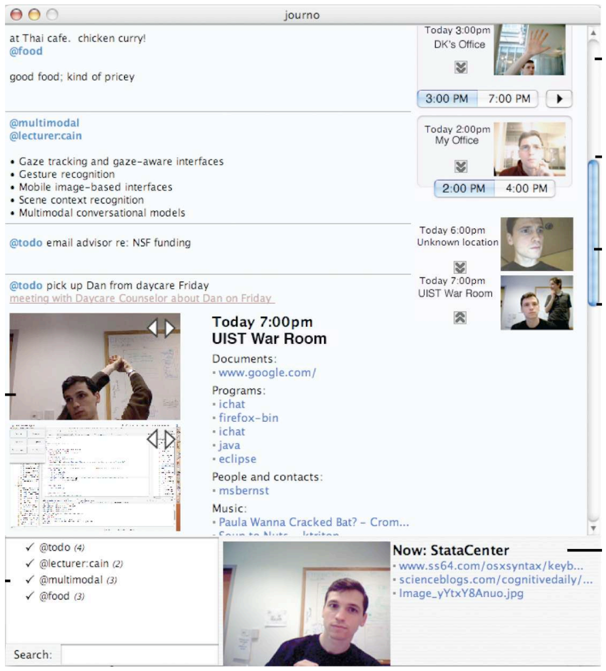
Reminding. Many participants took advantage of information scraps' visibility and mobility by placing them in the way of their future movements to create reminders for themselves. Participants used techniques such as colored Post-its or unread or unfiled e-mails, reminding them to take action later.

Unusual Data Types. Taking advantage of information scraps' freeform nature, participants managed unique data types that might have otherwise remained unorganized. For example, one participant created an information scrap system to manage a library-style checkout for his privately owned construction tools, and one participant maintained a complex document of contact information annotated with private notes on clients.

Information Scraps, PIM and IR

Information scraps present a challenging information retrieval task. It is appropriate to consider IR with respect to scraps' lifecycle because of the sheer number of scraps our participants compiled, in tension with the need to re-find specific scraps later. However, information scraps do not easily lend themselves to traditional IR approaches. Scraps are often recorded incompletely, written tersely, or intentionally left ambiguous – making it more difficult for IR algorithms to parse the content. Conversely, the user may recall the content via a completely different set of cues than the content itself: for example, context surrounding note creation ("I wrote it while in the elevator.") or gestalt meaning ("My notes from that meeting about funding.").

Depending on the role an information scrap is playing, its information retrieval needs may further vary. For example, work-in-progress scraps may often contain very little explicit information to index – for example, consider a notebook page full of rough interface sketches. It is also questionable whether traditional IR metrics and tasks are even



at Thai cafe. chicken curry!
@food

good food; kind of pricey

@multimodal
@lecturer:cain

- Gaze tracking and gaze-aware interfaces
- Gesture recognition
- Mobile image-based interfaces
- Scene context recognition
- Multimodal conversational models

@todo email advisor re: NSF funding

@todo pick up Dan from daycare Friday
meeting with Daycare Counselor about Dan on Friday

Today 7:00pm
UIST War Room

Documents:
- www.google.com/

Programs:
- ichat
- firefox-bin
- ichat
- java
- eclipse

People and contacts:
- msbernst

Music:
- Paula Wanna Cracked Bat? – Crom...

Today 3:00pm
DK's Office

3:00 PM 7:00 PM

Today 2:00pm
My Office

2:00 PM 4:00 PM

Today 6:00pm
Unknown location

Today 7:00pm
UIST War Room

✓ @todo (4)
✓ @lecturer:cain (2)
✓ @multimodal (3)
✓ @food (3)

Search:

Now: StataCenter
- www.ss64.com/osxsyntax/keyb...
- scienceblogs.com/cognitivedaily/...
- Image_yYtxY8Anuo.jpg

Figure 2. Jourknow, our prototype information scrap management tool.

appropriate for information scraps. Reminder scraps, for example, are intended to proactively remind rather than be indexed, searched or visualized later, and temporary storage scraps's relevance to the user decreases quickly as the scraps age.

JOURKNOW: A NEW INFORMATION SCRAP TOOL

In parallel with our ethnographic study, we have been designing an information scrap management tool (Figure 2) [2]. We briefly mention Jourknow, our information scrap client, which focuses on the following ideas:

Structure Extraction. In order to elevate ambiguous or unstructured text to searchable, sortable data, we can assist in the conversion of the raw scrap rich in implicit structure to data with explicit metadata structure. Thus, "mtg. w/ Karger @ 5" becomes reified as a calendar event in the user's calendar application and searchable as such.

Context Association. We can further assist in information scrap retrieval tasks by allowing the user to query by the situation surrounding the note capture. Jourknow automatically captures and associates information surrounding the user's situation. This data includes day and time, location hints (e.g., wifi ssid), events scheduled on the calendar, and activity traces including web pages, active applications, people the user communicated with, and pictures of the desktop and the user. Users may then use a faceted browsing interface to search for notes fulfilling specific criteria.

Mobility. We are constructing a mobile version of the Jourknow client to run on users' cellular phones. The mobile version of the client is intended to support users when away from the computer, tailored to the information capture needs when mobile and affordances the cell phone offers.

We recently completed a weeklong deployment evaluation of the Jourknow client to help determine its strengths and weaknesses in information scrap management.

CONCLUSION

In this position paper we have raised the information retrieval problem as it affects and is affected by personal information management. We examined information scraps as a particularly interesting case of personal information in this respect, considering features which might bear on IR tasks. We have seen that importing traditional IR goals and metrics into the space of information scraps fails to account for many of the distinct qualities of scraps' encoding and usual retrieval cues. We report on our efforts to bridge this gap via our prototype system Jourknow, which attempts to make easier the capture and retrieval of information scraps.

REFERENCES

1. M. Bernstein, M. V. Kleek, D. Karger, and mc schraefel. Information scraps: How and why information eludes our personal information management tools. *In Submission to ACM Transactions on Information Systems*, 2007.

2. M. V. Kleek, M. Bernstein, D. Karger, and mc schraefel. Gui? – phooey! the case for text input. In *Proc. UIST '07*, September 2007.

Collaborative Exploratory Search

Jeremy Pickens and Gene Golovchinsky

HCIR 2007
Workshop Boston

INTRODUCTION

Most modern information retrieval (search) systems are geared toward helping a user quickly and effectively find a particular item. That item may be a document, a geographic location, a factoid, etc. This approach is good when a single piece of information can fulfill the user's information need.

In many situations, however, multiple items and/or richer overviews of the entire information space are necessary. In support of this, exploratory search systems have been developed. Exploratory search systems typically blend a variety of information seeking tools and tactics (e.g., querying, browsing, document clustering, etc.) to help the user better understand the range of available information. Such tactics are combined in opportunistic ways as users' understanding of their information needs evolves [Bates, 1989],

Currently, there is also much work around mass social collaboration, or aggregation of large-scale user intention and information seeking behaviors. Recommender systems such as Amazon [cite, Glinden], personalization systems based on behavioral or topical clustering such as Kaltix (Google) [cite, 2003], community-mediated search collaboration such as iSpy [cite Barry Smyth] and bookmarking and voting sites such as del.icio.us and Digg all make use of the "wisdom of crowds" approach to collaboration. The underlying commonality among these systems is the notion of using correlated aggregate behavior to steer the individual searcher toward the most relevant pieces of information based on results obtained by others. And individual, working alone, is implicitly informed by prior search behaviors of the crowd.

But what if a searcher is looking for novel information, or for information that does not have a large peer group from which to draw recommendations? What if the searcher's goals are different from others who formulated similar search requests? The problem with the crowd-based approaches is two-fold: there may be large numbers of documents in a system with no prior user attention, and the information need of the crowd might not match the need of the current searcher. These disparities may limit the effectiveness of the otherwise desirable strategy of using multiple people's input to determine the outcome of a search session.

We propose to mitigate the deficiencies of correlated search with collaborative search, that is, search in which a small group of people shares a common information need and actively (and synchronously) collaborates to achieve it. Furthermore, we propose a system architecture that mediates search activity of multiple people by combining their inputs and by specializing results delivered to them to take advantage of their skills and knowledge.

For example, the goal of previous crowd-based systems is to help the current searcher discover relevant items that others have already discovered. Interesting technological challenges involve discovering useful crowd clusters or latent spaces, figuring out which prior users' actions are relevant the current user's information need, etc. In an explicitly collaborative, small team-based environment, on the other hand, no guesswork is needed for user search behavior clustering; one's collaborators are known. The focus of the system shifts from helping individuals re-find information that has already been found by someone else to helping a team member find information that no other member has yet found, but that is relevant to the overall information needs. Explicit collaboration therefore allows a fundamental conceptual shift in system design, away from algorithms and interfaces that support re-finding and re-discovering to algorithms and interfaces that support new discovery and exploration.

	Explicit	Implicit
synchronous	Collaborative exploratory search (FXPAL)	Realtime awareness and continual-update context systems (e.g. Imity, or)
asynchronous	"search Trails" [1]	Web 2.0, (recommender systems, Google's Kaltix, wisdom of crowds)

Table 1. Collaborative Exploration Search design space.

THE DESIGN SPACE

We can characterize the design space along two dimensions, as shown in Table 1. Search can be explicit or implicit, asynchronous or synchronous. The main difference between explicit and implict collaboration is whether people to work toward the same goal (explicit) or similar but independent goals (implicit). The key difference between synchronous and asynchronous collaboration is that synchronous collaboration occurs in a tight real (or near-real) time feedback loop, whereas asynchronous collaboration lacks that immediate interaction.

These key differences imply the need for different user interfaces and interaction styles, and pose interesting design challenges not only for the underlying architecture but also for the interfaces with which people interact.

One key advantage of small teams is the breadth of experience they bring to the search task. The nature of this experience may be expressed as roles. In some cases, each team member plays a similar role, whereas in other cases, the roles are different.

ROLES

There are a number of areas in which collaborative exploratory search of this nature is useful. Here are two examples.

- Domain expert/domain expert: two doctors from different domains need to collaborate on a complex diagnostic question, involving esoteric information from each domain. Each generates domain-specific search terms that the system integrates into joined queries. Results are allocated to each user based on domain expertise.
- Domain expert/search expert collaboration: the collaborators are a domain expert (aerospace engineer) and a search expert (librarian). The system allows the librarian to select sources and to formulate queries, while the engineer suggests terms and provides relevance feedback.

HUMAN-COMPUTER INTERACTION

A system that supports synchronous collaborative search must be able to collect data from two or more people and to use that data to synthesize queries and other operations on a search engine, and it must allocate search results to participants as appropriate to their roles.

Such a system must include interfaces for individual exploration, interfaces that support implicit awareness of others' activities, and interfaces for explicit collaboration. A middleware layer for coordinating the activities of the group, and an algorithmic engine optimized for collaborative exploratory search must exist to support this rich information sharing.

This system architecture can allow a small group of focused information seekers to search collections in concert. The system provides feedback based not only on an individual's search behavior, but also on the current, active search behavior of one's search allies.

User interface design in such an environment must draw on findings from CSCW, HCI, and IR fields.

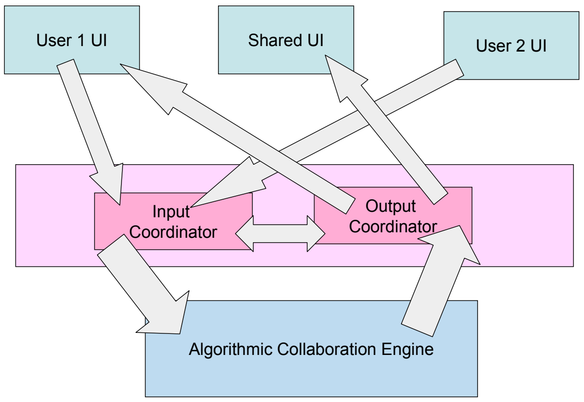


Figure 1. System architecture

EVALUATION

It is desirable to evaluate collaborative information seeking in traditional IR terms, but care must be taken when comparing search results of systems with different numbers of users working synchronously. One possible approach is to produce a combined ranked list based on contributions of all searchers, and then compute mean average precision (MAP) scores that are adjusted to account for the number of people contributing to the results.

One possible way to adjust MAP scores is to penalize highly-ranked non-relevant documents based on the number of searchers. The logic is based on the assumption that identical results produced by a smaller team are better. Thus the more people fail to make a correct judgment about a document, the worse that team performs.

$$modifiedprecision = \frac{|relevant \wedge retrieved|}{|relevant \wedge retrieved| + \left(|\overline{relevant} \wedge retrieved| * f(S)\right)}$$

Where f(S) is the penality function, such that f(1) = 1 and f(S>1) > 1. While a number of such functions are possible, we suggest that f(S)=S is a good place to start. The exact choice of function will depend on a variety of factors, and we need much more experience with these functions to determine which are appropriate when.

CONCLUSIONS

REFERENCES

1. Chi, E.H., Pirolli, P., Chen, K. and Pitkow, J, Using information scent to model user information needs and actions and the Web. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, (Seattle, WA, United States, 2001), ACM Press, 490-497.
2. Morris, M.R., Interfaces for Collaborative Exploratory Web Search: Motivations and Directions for Multi-User Designs. In Proceedings of the CHI 2007 Workshop on Exploratory Search and HCI., (2007).

Codifier: A Programmer-Centric Search User Interface

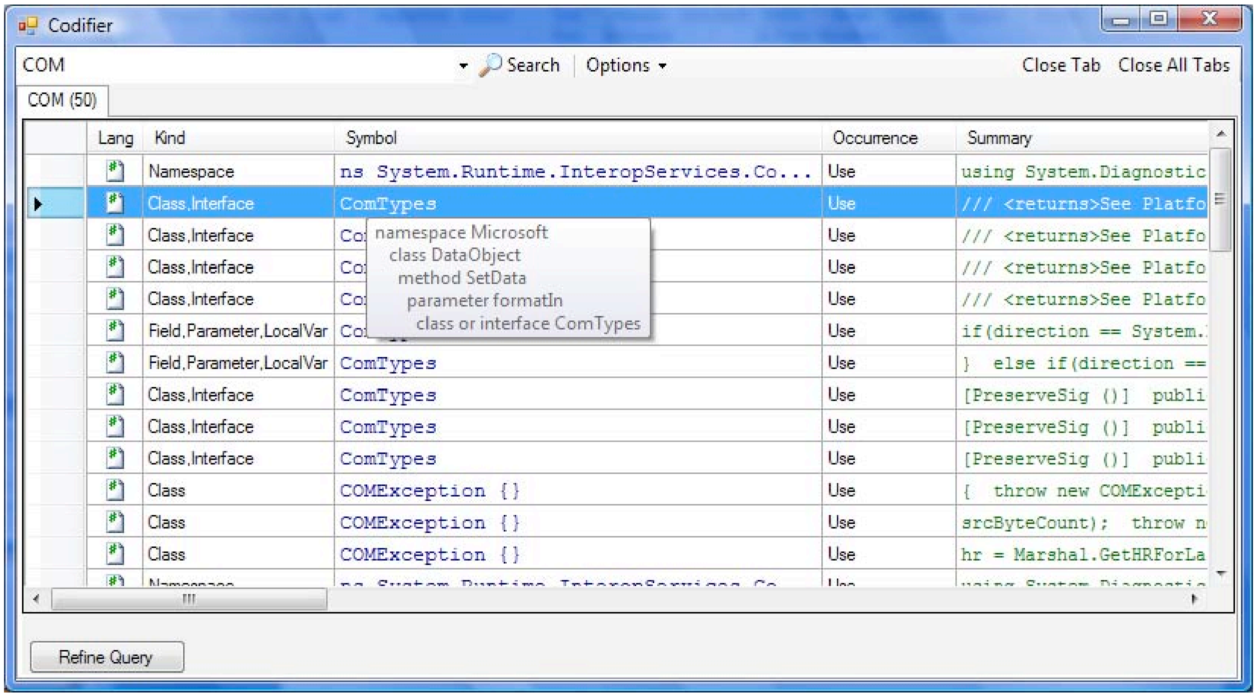
Andrew Begel
Microsoft Research
Redmond, WA 98052
andrew.begel@microsoft.com

Search tools have transformed knowledge discovery by exposing information from previously hidden repositories to the workers who need it. Search engines like Google and Live.com provide search capabilities via a simple one-line text query box, and present results in a paged HTML list. When the repository being searched contains structured information with extractable metadata (e.g. program source code), it can be advantageous to index the metadata and use it to enable queries that are more task-centric and suitable for an domain-specific audience.

Codifier is a programmer-centric search user interface that enables software developers to ask domain-specific questions related to programming languages and software. For example, developers might ask

- Where is this API or data structure defined?
- Where is this API used?
- Where is this variable assigned a value?
- I know this function writes data to the disk, but I forget exactly what its name is.
- Even if I spell it wrong, I still want to find IPersistentItemsChangedSink.
- Find all functions where Open() is called with Init().
- Show me all calls to this method, so I can refactor it by hand.

We index C, C++, C# and VBScript program source code using a modified compiler to extract and store lexical and syntactic metadata into a SQL Server 2005 or Windows Desktop Search database. The Codifier user interface, presented in Figure 1, enables software developers to search in this database for symbols found anywhere in the indexed source code (not just their definitions). Searches supported by metadata can be quite powerful. In additional to source code symbols, we can search for language-specific connectors (e.g. `foo::bar`, `foo.bar`,



Lang	Kind	Symbol	Occurrence	Summary
	Namespace	`ns System.Runtime.InteropServices.Co...`	Use	`using System.Diagnostic`
	Class,Interface	`ComTypes`	Use	`/// <returns>See Platfo`
	Class,Interface	`Co`	Use	`/// <returns>See Platfo`
	Class,Interface	`Co`	Use	`/// <returns>See Platfo`
	Class,Interface	`Co`	Use	`/// <returns>See Platfo`
	Field,Parameter,LocalVar	`Co`	Use	`if(direction == System.`
	Field,Parameter,LocalVar	`ComTypes`	Use	`} else if(direction ==`
	Class,Interface	`ComTypes`	Use	`[PreserveSig ()] publi`
	Class,Interface	`ComTypes`	Use	`[PreserveSig ()] publi`
	Class,Interface	`ComTypes`	Use	`[PreserveSig ()] publi`
	Class	`COMException {}`	Use	`{ throw new COMExcepti`
	Class	`COMException {}`	Use	`srcByteCount); throw n`
	Class	`COMException {}`	Use	`hr = Marshal.GetHRForLa`

```
namespace Microsoft
  class DataObject
    method SetData
      parameter formatIn
        class or interface ComTypes
```

Figure 1: Codifier search user interface showing a search for the symbol COM.

`foo->bar`), synonyms, homophones, abbreviations, concept keywords (e.g. searching for COM finds `COMString`, `ComException`, `ICOMPointer`), kind and usage of symbols (e.g. searching for definitions of methods named `WriteString` (kind:method usage:def WriteString), newly instantiated objects of class `IEnumString` (kind:class usage:use IEnumString), assignments to local variables named `firstTimeThroughLoop` (kind:localvar usage:assign firstTimeThroughLoop)), lexical scoping (e.g. searching for calls to method `Open()` in classes named `MemoryAccess`), and keywords for searching by programming language, source control information and file path.

Filtering, sorting and refinement capabilities are important for winnowing the thousands of answers resulting from a search over a large source code base. For example, Microsoft Windows 2003 Server contains several hundred million symbols in its source code – when searching for code, the "right" result may be one out of thousands, or may be *all* of them. Codifier provides support for filtering results based on the lexical, syntactic, and file path scope of the result. In addition, by presenting the results in a grid, with one row per symbol found, the UI enables sorting based on any of the metadata values. Refinement of queries is supported by metadata facet. A top 10 list of results is shown for each facet. When the user clicks on one of them, an additional filtering term is added to the query, which is then re-executed.

Codifier stores one symbol per row when using SQL Server 2005. Using as-of-yet unoptimized schema, each symbol's metadata is stored in about 2,300 bytes of space on disk, so even the largest bodies of source code we index fit into less than 500 GB. Indexing time is about 2 million symbols per hour. When using the less scalable Windows Desktop Search 3.0, Codifier stores all metadata for the symbols in each file in the inverted index, enabling metadata-based searches, but requiring reanalysis and extraction of metadata when each search result is retrieved. Indexing time with WDS is about 80,000 files per hour.

Other search engines such as Google Code Search, Krugle.com, and Koders.com have been targeted at program source code, but these index minimal metadata, and mostly function by restricting the scope of full-text search to source code files. Numerous IDEs such as Visual Studio and Eclipse support symbolic searches with full metadata support, but have limitations as well. Eclipse has a GUI-based interface to metadata specification which can be onerous to enter, and both IDEs limit searches to a single managed project at a time. The Source Insight IDE supports a larger search scope using heuristically-evaluated metadata, but does not support synonyms, homophones, concept keywords or lexical scoping in queries or results. Various research projects such as GENOA [4], SCRUPLE [7], TAWK [1], and a project by Clarke, Cox and Sim [3], have emphasized the back-end techniques of indexing code and left their front ends to technical pattern-matching languages. Strathcona [6] searches for code by example, alleviating the query language problem. Sourcerer [2] and Assieme [5] search for links within public code (and Assieme links in web-based descriptions) to enable programmers to learn how to use new APIs. We have made Codifier's query language straightforward, like a typical web search engine, and concentrate mainly on improving the usability of the UI for understanding and manipulating search results.

Codifier will be demoed at the workshop and comments and feedback will be gratefully appreciated.

[1] Atkinson, D. C. and Griswold, W. 2006. Effective pattern matching of source code using abstract syntax patterns. *Softw. Pract. Exper.* 36, 4 (Apr 2006), 413-447.

[2] Bajracharya, S., Ngo, T., Linstead, E., Dou, Y., Rigor, P., Baldi, P., and Lopes, C. 2006. Sourcerer: a search engine for open source code supporting structure-based search. In *Companion To OOPSLA.* (Portland, Oregon, USA, Oct 22 - 26, 2006). ACM Press, 681-682.

[3] Clarke, C., Cox, A., and Sim, S. 1999. Searching program source code with a structured text retrieval system (poster abstract). In *Proceedings of SIGIR.* (Berkeley, California, United States, Aug 15 - 19, 1999). ACM Press, 307-308.

[4] Devanbu, P. T. 1992. GENOA: a customizable language- and front-end independent code analyzer. In *Proceedings of ICSE.* (Melbourne, Australia, May 11 - 15, 1992). ACM Press, 307-317.

[5] Hoffman, R., Fogarty, J., and Weld, D. Assieme: Finding and Leveraging Implicit References in a Web Search Interface for Programmers. To appear in *Proceedings of UIST.* (Newport, Rhode Island, Oct 7-10, 2007). ACM Press.

[6] Holmes, R., Walker, R. J., and Murphy, G. C. 2005. Strathcona example recommendation tool. In *Proceedings of ESEC.* (Lisbon, Portugal, Sept 05 - 09, 2005). ACM Press, 237-240.

[7] Paul, S. and Prakash, A. 1994. A Framework for Source Code Search Using Program Patterns. *IEEE Trans. Softw. Eng.* 20, 6 (Jun. 1994), 463-475.

Authority facets: Judging trust in unfamiliar content

Peter Bell
Endeca

How do people judge trust in content that has no provenance, like the stamp of an editorial process or audit trail? Most digital content lacks the authority of traditional publication models, and in content created through social collaboration, like the Wikipedia, it even lacks a clear single author. Nevertheless, informal content is widely consumed, albeit with healthy skepticism.

Trust and authority are not binary — present or absent. In fact, trust in a source varies depending on a user's task. For example, a financial analyst might:

- Trust numbers reported in *The Economist* enough to prepare a report that will be filed with the SEC.
- Disagree with the editorial opinion of an Economist blog on tax law reforms, but will follow links to sources provided by the author to research more.
- Might not trust an anonymous posting on an Economist message board, yet finds a tip worthy of emailing the author for further due diligence.

In each case, content with widely varying degrees of authority proved to be helpful, given proper context.

People judge trust for themselves constantly, and rely on nuanced evidence to do so. In a faceted navigation system it is possible to supply readers with abundant evidence to determine trust for themselves through "authority facets." Just as facets help users filter and browse content by subject or attributes, authority facets can add rich information about the trustworthiness of content by an author for a given task. Examples include facets on skills, ratings, certifications, affiliations, and social network ties.

There are at least two approaches to implementing authority facets:

- Content is tagged with authority facets and users can navigate based on those tags. Example: at an ecommerce site with user-generated product reviews, users can navigate to camera reviews written just by novices, intermediates, or experts. A review stating a camera is "heavy" might be judged to carry different meanings coming from novice vs. expert authors.
- Facets themselves can have authority facets, and users can filter the values in a facet by its authority facets. Example: the author facet in a university Wiki is associated with its own facets, like department, seniority, and certifications. A student searching for comments in a blog might narrow the author facet to find just authors that are on faculty, and then use that subset to filter blog comments.

Authority facets are just beginning to emerge in faceted navigation systems, and show promise as a way to enrich content so users can determine its suitability for a given task. This brief survey of examples will show how authority facets are being used today and suggest future directions.

Mapping the Design Space of Faceted Search Interfaces

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

Introduction

Faceted search, guided search, and categorized overviews are becoming accepted techniques to support complex information seeking tasks like exploratory search. There are a growing number of applications that use these techniques for library catalogs, web search, shopping, image collections, and other domains (Antelman, Lynema, & Pace, 2006; Hearst et al., 2002; Tunkelang, 2006; Yee, Swearingen, Li, & Hearst, 2003). Design guidelines for the application of these techniques are starting to emerge (Hearst, 2006; Kules & Shneiderman, to appear), but there is no systematic description of the design space for faceted search interfaces. An understanding of the design space will aid designers by alerting them to design options and decisions they should address. It will aid researchers by suggesting a framework for guidelines as well as additional areas of study. In particular, it may help understand the actions, tactics and strategems (Bates, 1990) supported by faceted search interfaces.

The objective of this paper is to begin identifying and structuring a set of dimensions of the design space for categorized overviews of search results. This paper proposes a set of dimensions for the design space of faceted search interfaces and two structures for meaningfully organizing them. These dimensions and the organizing structures emerged from analysis of recent literature and applications from several domains.

Method

We began with design dimensions extracted from Hearst (2006), Smith & Kules (2006), and Kules (2006). Hearst (2006) makes detailed design recommendations for hierarchical faceted search interfaces. Kules (2006) identifies a set of ten dimensions and their corresponding design options for categorized overviews. Smith & Kules (2006) proposes 14 dimensions in three areas (organization, display, and interaction). We extend those 14 dimensions by analyzing additional interfaces from different domains: mSpace (schraefel et al., 2005), the Relation Browser (Marchionini & Brunk, 2003), and several commercial shopping interfaces.

This analysis yielded 28 dimensions. By considering the three primary conceptual elements (the facets, the categories within the facets, and the individual search result items), we structure the interactions by examining how an action on each element can affect that element and the other two. Actions include, but are not limited to, clicking on, dragging, and "hovering" over an interface element. For example, clicking on a category in a faceted search interface often affects the items (by narrowing the set of results to that category) and the categories displayed (by displaying the subcategories). Structuring the design dimensions in this manner yields two tables. Table 1 contains

Table 1. Design dimensions related to the organization and display of facets, categories, and items.

	Organization	Display
Facet	• Ordering • Grouping • Semantics/relationship represented	• Location & layout • Method for determining which facets are displayed (e.g., predetermined, user-selected) • Form of display (textual or graphical) • Display of facets with 0 or 1 non-empty categories (e.g., no change, shrink or hide facet)
Category	• Ordering • Breadth & depth • Labels & terms • Categorization method (e.g., use existing metadata, extract or infer categories, automated clustering)	• Visible levels of hierarchy • Method of determining which categories to display (e.g. all, most common, show/hide empty categories, provision of keyword search/filter on category name) • Sorting/grouping of displayed categories • Display of an "Uncategorized" pseudo-category for uncategorized items
Item	• Ordering	• Method of determining which items to display (e.g. display N per page, display a sample for each visible category)

design dimensions related to the organization and display of facets, categories, and items. Table 2 structures the interaction dimensions into a 3x3 array. The rows represent the element being acted upon and the columns represent the element being affected. The empty cells suggest areas for additional study. One remaining dimension does not fit in these structures: Breadcrumbs (how they are ordered; the effect of removing an element of the breadcrumb list).

Conclusion

The organization and dimensions described here are a work in progress, intended to stimulate discussion. This is one step in developing an understanding of the design space, starting with the current literature and a sample of applications targeted at desktop-based web browsers. Additional actions, interactions, and dimensions will certainly emerge from the study of other applications and non-desktop devices (e.g. PDAs).

Table 2. Design dimensions related to the interaction of facets, categories, and items. The rows contain the elements being acted upon by the user and the columns contain the elements being affected.

		Affected element		
		Facet	**Category**	**Item**
Element being acted upon by user	**Facet**	• Selection (simultaneous or sequential)		
	Category	• Effect of category selection on other facets (e.g., display subcategories as a new pseudo-facet)	• Effect of category selection on display of other categories within facet (e.g. are multiple selections supported) • Previews of subcategories	• Narrowing, broadening • Sorting/grouping items (e.g., group by children of most recently selected category) • Previews
	Item	• Highlight related facet	• Highlight category membership	• Find related items (e.g. "More like this"

References

Antelman, K., Lynema, E., & Pace, A. (2006). Toward a 21st Century Library Catalog. *Information Technology and Libraries, 25*(3), 128-139.

Bates, M. (1990). Where should the person stop and the information search interface start. *Information Processing and Management, 26*(5), 575-591.

Hearst, M. (2006). *Design recommendations for hierarchical faceted search interfaces*. Paper presented at the ACM SIGIR 2006 Workshop on Faceted Search. Retrieved September 27, 2007, from http://flamenco.berkeley.edu/papers/faceted-workshop06.pdf.

Hearst, M., Elliot, A., English, J., Sinha, R., Swearingen, K., & Yee, P. (2002). Finding the flow in web site search. *Communications of the ACM, 45*(9), 42-49.

Kules, B. (2006). *Supporting Exploratory Web Search with Meaningful and Stable Categorized Overviews* (Unpublished doctoral dissertation): University of Maryland, College Park. Retrieved May 30, 2007, from http://hcil.cs.umd.edu/trs/2006-14/2006-14.pdf.

Kules, B., & Shneiderman, B. (to appear). Users can change their web search tactics: Design guidelines for categorized overviews. *Information Processing & Management*.

Marchionini, G., & Brunk, B. (2003). Towards a General Relation Browser: A GUI for Information Architects. *Journal of Digital Information, 4*(1), Article No. 179.

schraefel, m. c., Smith, D. A., Owens, A., Russell, A., Harris, C., & Wilson, M. (2005). *The evolving mSpace platform: Leveraging the semantic web on the trail of the memex*. Paper presented at the Proceedings of the Sixteenth ACM Conference on Hypertext and Hypermedia.

Smith, J., & Kules, B. (2006). *Toward a design space for categorized overviews of search results*. Retrieved September 27, 2007, from http://faculty.cua.edu/kules/Papers/SmithKules_DesignSpace.pdf.

Tunkelang, D. (2006). *Dynamic Category Sets: an approach for faceted search*. Paper presented at the ACM SIGIR 2006 Workshop on Faceted Search. Retrieved September 27, 2007, from http://www.cs.cmu.edu/~quixote/DynamicCategorySets.pdf.

Yee, K.-P., Swearingen, K., Li, K., & Hearst, M. (2003). Faceted metadata for image search and browsing. In *Proceedings of the SIGCHI Conference on Human factors in Computing Systems, Ft. Lauderdale, FL* (pp. 401-408). New York: ACM Press.

Images as Supportive Elements for Search

Giridhar Kumaran and Xiaobing Xue
Center for Intelligent Information Retrieval, Department of Computer Science
University of Massachusetts Amherst, Amherst, MA 01002
{giridhar, xuexb}@cs.umass.edu

Introduction

The dominant paradigm of search today is heavily biased towards textual interfaces. Users enter textual queries, and navigate to potentially relevant content guided by short textual snippets offering summaries of retrieved information. This interaction paradigm is not only quite successful in practice but also provides an opportunity for improved techniques that are potentially even easier and effective. Our work focuses on that part of the interactive retrieval process where users are offered textual cues to guide them towards relevant content. By using images in lieu of text, we believe we can provide a user experience that is not only more effective, but also more efficient.

Images as supportive elements.

A study reported in (Coltheart, 1999) observed that a person can get the gist of an image in 110ms or less while in the same time she can only read less than 1 word, or skim 2 words. This was the basis for previous research (Xue et al, 2006) in the web domain that showed that using images in conjunction with text improves user experience and satisfaction. Usually, web documents have images included in them, making the task of finding appropriate supportive images easier. We are interested in extending this idea to collections where documents do not have associated images. We believe that this is important as vast amounts of information in historical archives, corporate intranets, scanned books etc. do not have images associated with them.

Our proposed approach is to build on standard information retrieval techniques and available resources like image search APIs to bring the same advantages of image-supported search to the collections we are interested in. We envision a procedure starting with retrieval of a set of documents from the collection in response to a user's query. The next step is to cluster the retrieved documents. Once the clusters are created we propose to create concise textual summaries representing each cluster, and using those summaries as queries for image search using an image search API. Alternately, we propose to search the web for documents similar to the cluster centroids, and use the images associated with them as supportive images for the user to work with.

We hypothesize that such image-supported search will not only improve precision, but also recall since the user can quickly sift though the images summarizing the ranked list, indirectly accessing documents further down.

References

Coltheart, V., ed. 1999. *Fleeting Memories: Cognition of Brief Visual Stimuli*. Cambridge, MA: MIT Press.

X.-B. Xue, Z.-H. Zhou, and Z. Zhang. *Improve web search using image snippets*. In: Proceedings of the 21st National Conference on Artificial Intelligence (AAAI'06), Boston, MA, 2006, pp.1431-1436.

Searching Conversational Speech

Mark Maybury
Information Technology Center
The MITRE Corporation
202 Burlington Road
Bedford, MA 01730, USA
maybury@mitre.org
itc.mitre.org

ABSTRACT
This paper summarizes two MITRE efforts to address the speech search challenge. We first describe Audio Hot Spotting (AHS) and then Cross Language Automated Speech Recognition (CLASR).

HUMAN LANGUAGE TECHNOLOGY AT MITRE

MITRE has engaged in a highly diverse HLT program over multiple decades. This has resulted in operational systems of integrated capabilities such as the DARPA MITRE Text and Audio Processing System (MiTAP) and the Translingual Instant Messaging (TrIM). MITRE has made a number of its contributions available via open source including:

- DARPA Galaxy Communicator architecture (~800 downloads at communicator.sourceforge.net)
- Midiki MITRE dialog manager toolkit (200+ downloads at midiki.sourceforge.net)
- Callisto annotation tool framework (~900 downloads at callisto.mitre.org)

We have also been active in facilitating the community to advance a number of key standards such as TIMEX2 (Ferro et al 2005), TimeML (timeml.org, Pustejovsky, et al. 2005), and more recently an effort to create SpatialML (2007). We have been awarded a patent for our effort in Broadcast News Navigation (US Patent 6,961,954 ; Maybury et al 1997) and have patents submitted for Personalcasting and Audio Hot Spotting and have advocated advanced Question Answering (Maybury 2004).

AUDIO HOT SPOTTING

The Audio Hot Spotting project (Hu et al. 2004) aims to support natural querying of audio and video, including meetings, news broadcasts, telephone conversations, and tactical communications/surveillance. As Figure 1 illustrates, the architecture of AHS integrates a variety of technologies including speaker ID, language ID, non speech audio detection, keyword spotting, transcription, prosodic feature and speech rate detection (e.g., for speaker emotional detection), and cross language search.

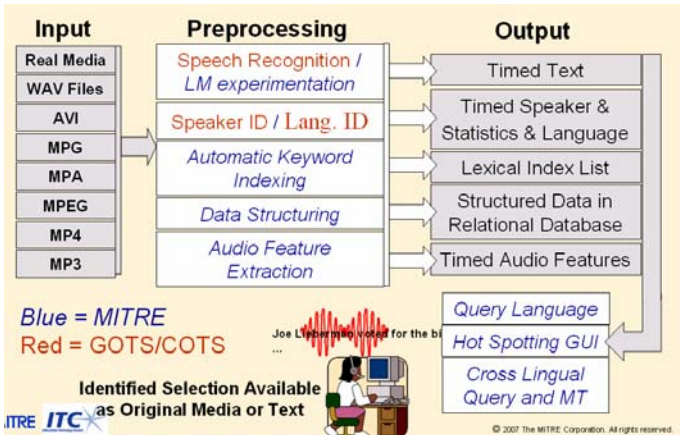

Input
Preprocessing
Output
Real Media
WAV Files
AVI
MPG
MPA
MPEG
MP4
MP3
Speech Recognition / LM experimentation
Speaker ID / Lang. ID
Automatic Keyword Indexing
Data Structuring
Audio Feature Extraction
Timed Text
Timed Speaker & Statistics & Language
Lexical Index List
Structured Data in Relational Database
Timed Audio Features
Query Language
Hot Spotting GUI
Cross Lingual Query and MT
Blue = MITRE
Red = GOTS/COTS
Joe Lieberman voted for the bi
Identified Selection Available as Original Media or Text
ITC
© 2007 The MITRE Corporation. All rights reserved.

Figure 1. AHS Architecture

An important innovation of AHS is the combination of word-based speech recognition with phoneme-based audio retrieval for mutual compensation for keyword queries. Phoneme-based audio retrieval is fast, more robust to spelling variations and audio quality, and may have more false positives for short-word queries. In addition, phoneme-based engines can retrieve proper names or words not in the dictionary (e.g., "Shengzhen") but, unfortunately, produces no transcripts for downstream processes. In contrast, word-based retrieval is more precise for single-word queries in good quality audio and provides transcripts for automatic downstream processes. Of course it has its limitations too. For example, it may miss hits for phrasal queries, out-of-vocabulary words, and in noisy audio and is slower in preprocessing.

Figure 2 illustrates the user interface for speech search, and includes a speaker and keyword search facility against both video and audio collections. The user can also search by non speech audio (e.g., clapping, laughter). A recent extension enables a user to query in English, have this query translated to a foreign language (e.g., Spanish, Arabic), use this query to retrieve hot spots in a transcription of the target media, which is then retrieved and translated into the query language.

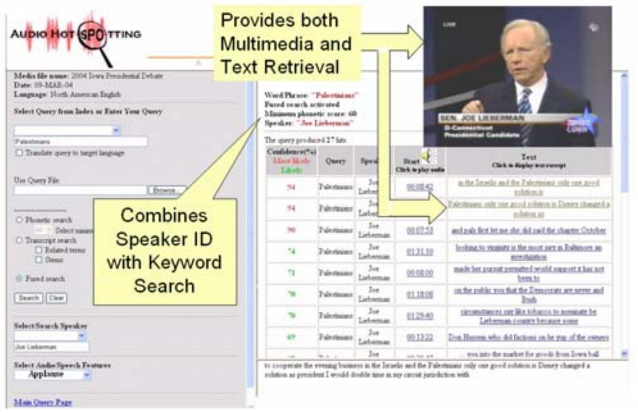

Audio Hot Spotting
Provides both Multimedia and Text Retrieval
Combines Speaker ID with Keyword Search
SEN. JOE LIEBERMAN
Browse
Search
Clear
Main Query Page

Figure 2. AHS Search Interface

CROSS LANGUAGE ASR (CLASR)

Access to foreign language spoken discourse is challenging. Building systems to do so is even more difficult when no written resources for that language exist. The Cross Language Automated Speech Recognition (CLASR) effort investigates a new approach for spoken language translation of languages that lack significant written resources. This effort is exploring the hypothesis that recent advances in both speech recognition and machine translation enable a fresh approach.

In particular, CLASR aims to build a process that goes from audio in a foreign language to text in English, addressing languages that do not have the right quantity and type of language resources for the current approaches. Current approaches to this challenge go from source language acoustics to source language written form, then from the source language written form to the English written form. Typically they use 1-best ASR output although some use n-best, but in all cases they output written form. CLASR simplifies this process and folds the translation model and acoustic model into one cross-language acoustic model.

While CLASR aims to address low resource languages, experiments are being performed on well-known languages (Spanish and Mandarin) to compare the new single stage approach to the traditional two stage pipe-line system, i.e., ASR+MT. In particular, CLASR uses an open source toolkit for ASR (HTK from Cambridge University) and a development kit for MT (GIZA++ and PHARAOH (JHU, MIT)). Our Spanish experiments are based on 30 hours of broadcast news audio using audio from Central America and transcripts in Spanish which have been translated into English. Initial results with Spanish with no additional language model have been promising as assessed by BLEU (BiLingual Evaluation Understudy), i.e., the portion of 4-word sequences in MT output that are found in reference translations with a range from 0 (poor) to 100 (good). The very first single-stage score, an initial foothold as we begin hill climbing, was a BLEU score of 8. By contrast, the 2-stage ASR+MT scores achieved a word error rate of 45 and and a BLEU score of 13. Our recent system Spanish-English MT system has a BLEU score of 21, outperforming the two stage baseline.

In summary, this approach is analogous to the results reported in this workshop by Olsson (2007) in which a single, integrated model outperforms a sequence of transcription and retrieval. Notably, CLASR's combined approach shows promise both performance-wise as well as in terms of its limited requirement for language resources.

REFERENCES

1. Ferro, L., Gerber, L., Mani, I., Sundheim, B. and Wilson G. (2005) "TIDES 2005 Standard for the Annotation of Temporal Expressions" April 2005, Updated September 2005. http://timex2.mitre.org/annotation_guidelines/2005_timex2_standard_v1.1.pdf
2. Fiscus, J., Ajot, J., Garofolo, J. and Doddington, G. Results of the 2006 Spoken Term Detection Evaluation. 2007 SIGIR Workshop on Searching Spontaneous Conversational Speech, Amsterdam, 27 July 2007. p. 45-51.
3. Hu, Q., Goodman, F., Boykin, S., Fish, R., and Greiff, W. 2004. "Audio Hot Spotting and Retrieval Using Multiple Audio Features and Multiple ASR Engines". Rich Transcription 2004 Spring Meeting Recognition Workshop at ICASSP 2004, Montreal, Canada. http://www.nist.gov/speech/test_beds/mr_proj/icassp_program.html
4. Maybury, M. (ed.) 1997. *Intelligent Multimedia Information Retrieval*. Menlo Park: AAAI/MIT Press. (http://www.aaai.org:80/Press/Books/Maybury-2/)
5. Maybury, M. editor. 2004. *New Directions in Question Answering*. AAAI/MIT Press.
6. Olsson, S. Improved Measures for Predicting the Usefulness of Recognition Lattices in Ranked Utterance Retrieval. 2007 SIGIR Workshop on Searching Spontaneous Conversational Speech, Amsterdam, 27 July 2007. p. 1-5.
7. Pustejovsky, J., Ingria, B. Sauri, R., Castano, J., Littman, J., Gaizauskas, R., Setzer, A., Katz, G. and Mani, I. 2005. The Specification Language TimeML. In Mani, I., Pustejovsky, J. and Gaizauskas, R. (eds.), The Language of Time: A Reader, 545-557. Oxford University Press. http://timeml.org
8. SpatialML: Annotation Scheme for Marking Spatial Expressions in Natural Language, March 30, 2007. MITRE Technical Report. http://sourceforge.net/projects/spatialm

ACKNOWLEDGEMENTS

This paper includes a summary of the contributions of Qian Hu's AHS project and John Henderson's CLASR project and their associated technical teams.

Natural Language Access to Information for Mobile Users

Alexander Ran
Nokia Research Center Cambridge
3 Cambridge Center
Cambridge, MA 02142
Alexander.Ran@nokia.com

Raimondas Lencevicius
Nokia Research Center Cambridge
3 Cambridge Center
Cambridge, MA 02142
Raimondas.Lencevicius@nokia.com

Mobile devices store a rich set of structured information. The phone book application contains names, phone numbers, addresses and affiliations of personal contacts. The calendar application contains entries for meetings with participants, meeting location and time. Logs of dialed received calls are stored on the device as well as sent and received messages and emails.

Unfortunately, users can only access this information using specially designed applications that manage different subsets of this information. There are several significant problems with this situation.

The applications that manage data on a mobile device only provide a fixed and limited set of ways to access the information. The contacts in the phone book have affiliations with organizations, professions, titles, home and office addresses, and other attributes. However the only way you can access this information with current applications is using contact's name. There is no direct way to find answers for many reasonable questions like: Who do you know at Nokia? What is the office address of your lawyer? Who is the sales manager at AT&T store in Burlington? Although the information about your meeting includes subject, location and participants, the only way for you to access the meeting information is browsing it chronologically. There is no direct way to find answers for many reasonable questions like: When is your next meeting at MIT? Where do you meet John next week? Are you free on Wednesday afternoon in October?

Although data sets owned by different applications are semantically related, there is no simple way to make these relations explicit. You receive calls, exchange messages and have meetings with people in your address book. Places that you visit often correspond to addresses of people and organizations listed as your contacts and may appear as meeting places on your calendar. For an intuitive interaction with information the semantic relations between different data items need to be explicitly represented.

Clearly we need a better way to manage the information. But is this enough? Having a rich semantic repository that integrates all information into a semantic network is a significant step forward compared to arbitrarily partitioned, disconnected subsets of information, but it does not solve the essential problem of user interaction with complex information sets. It is the language barrier. The richer the information set we are dealing with, the richer language we need to interact with this information set. It seems plausible that the only general solution to this problem would involve some form of natural language based access to information.

Natural language interfaces to databases is not a new idea. Besides some inherent limitations of natural language interaction with a machine, a major factor for limited success of this technology was the fact that NLI were not easily portable between different databases. This is due to significant dependencies on data organization and content that had to be introduced in the language system component in order to generate database specific semantic representation of natural language request.

Let us assume the user asks the system about contacts in some organization and geographical location:

Who do I know at IBM Ulm?
Who are my contacts at IBM in Ulm?
What are the names of my contacts at IBM in Ulm?[1]

The operational semantics of these questions can be adequately represented with a database query. Let us consider how this request would need to be posed to an RDF [2] repository. SPARQL [3] query corresponding to our example question over our extended PIM ontology looks as follows:

```
SELECT DISTINCT ?person ?givenName ?familyName
FROM <http://localhost/pim.rdf>
WHERE {?person a pim:Person; pim:givenName ?givenName; pim:familyName ?familyName; pim:affiliation ?affiliation; pim:address ?person_address.
    ?affiliation pim:organization ?organization.
    ?organization pim:address ?organization_address; pim:name "IBM".
    {?person_address pim:locality "Ulm"} UNION
    {?organization_address pim:locality "Ulm"}}
```

Unfortunately in order for a language system to generate such semantic representation from the original questions, the language system must contain a large amount of information about the structure of the database and its content. Such information includes the facts that IBM is a name of an organization and Ulm is a name of a city, cities can be related to organization through their addresses, organizations are related to people through their affiliations, people are related to cities through their home and office addresses, and all these relationships and objects are represented by the specific structures and entities of the database.

Entering such information into a language system is a tedious and costly process that is not only domain dependent but also is sensitive to specific choices of database organization. However if it were possible to automate the integration of language system with a data repository the natural language interface proposition would become very attractive.

We are attempting to do exactly that, exploring the rules for design of semantic repositories that can be interpreted as a

[1] The name of the organization and the city were selected for shortness and carry no other information

grammar and a model for semantic interpretation of natural language requests concerned with access to information in the repository.

We have designed and implemented the Natural Query (NQ) language and engine [1] that accepts database independent semantic representation of natural language requests and using heuristics produces operational interpretation the natural language question over a given semantic repository. NQ requires attaching basic linguistic information to structural elements of semantic repository and imposes certain rules on the design of its ontology.

NQ relies on language tags being attached to database elements such as classes and properties. Multiple tags can be attached to a single element and a single tag can be attached to multiple elements. Language tags could correspond to the names of semantic categories used by the language system(s) or include them in their semantic class. Given a form like the one in our example,

contact.name: ?
organization: IBM
city: Ulm

NQ interprets it as:

find the attributes tagged as "name" of an instance of the class tagged as "contact" related through properties tagged as "organization" and "city" to values "IBM" and "Ulm" respectively

While a formal query defines a connected subgraph, the database independent meaning representation only identifies some nodes and edges of this subgraph. Identified fragment might be disconnected. In the example above it identifies "*Person*" and "*Organization*" classes as well as "*Ulm*" value of "*locality*" property (by reference to its language tag "*city*") and "*IBM*" as a value of "*name*" property of an instance of "*Organization*" class. This leads to an important idea: that the knowledge embedded in the formal queries that *know* the database organization can be also extracted from the natural language meaning representation and the data repository itself. For a given set of elements identified by a meaning representation of natural language request it is possible to identify the query subgraph by searching the database. In other words, a program could find paths connecting the nodes known from the meaning representation, such as *"Person", "name", "Organization", "City", "Ulm",* and *"IBM"*.

Therefore, while traditional approaches to semantic analysis of natural language questions over databases rely on hand crafted code or data for representing the information about the organization of the database, NQ extracts such knowledge from the data repository by using graph search. Given a question "*Who are my contacts at IBM in Ulm?*", NQ finds paths connecting the nodes known from the database independent meaning representation, such as *"Person", "name", "Organization", "City", "Ulm",* and *"IBM"*.

NQ may find multiple subgraphs that connect all given elements. In such cases we apply heuristic ranking of these subgraphs in order to determine the most relevant ones. So far we experimented with several ranking mechanisms all of which are variations on path length (weight) between the elements specified by the meaning representation. In all our experiments the results retrieved by the system in response to natural language questions correspond well with intuition of human subjects.

We are currently working on integrating NQ with the Galaxy natural language system [3] and exploring the potential of our approach to provide mobile users natural language access to semantic repositories and on and off the mobile device.

1. REFERENCES

[1] Ran, A., and Lencevicius, R., "Natural Language Query System for RDF Repositories", To appear in the Proceedings of International Symposium on Natural Language Processing, SNLP 2007.

[2] Resource Description Framework, http://www.w3.org/RDF/, 2007.

[3] S. Seneff, E. Hurley, R. Lau, C. Pao, P. Schmid, and V. Zue, "GALAXY-II: A Reference Architecture for Conversational System Development," *Proc. ICSLP 98*, Sydney, Australia, November 1998.

[4] SPARQL Query Language for RDF, http://www.w3.org/TR/rdf-sparql-query/, 2007

AnalogySpace and ConceptNet

Rob Speer and Catherine Havasi

October 19, 2007

When people communicate with each other, their conversation relies on many basic, unspoken assumptions, and they often learn the basis behind these assumptions long before they can write at all, much less write the text found in corpora. These assumptions underlie all forms of human communication, from teaching, to giving directions, to ordering dinner at a restaurant.

A user who interacts with a computer interface, however, can become frustrated because the computer does not understand their goals and motivations. For human-computer interaction to become as fluent as communication between humans, computers need to be able to understand the user's basic, unspoken assumptions. These assumptions form the body of knowledge known as "common sense".

Grice's theory of pragmatics states that when communicating, people tend not to provide information which is obvious or extraneous. If someone says "I bought groceries", he is unlikely to add that he used money to do so, unless the context made this fact surprising or questionable. Thus, it is difficult to collect common sense knowledge automatically from the Internet or a lexical resource.

Since 2000, the Open Mind Common Sense project has been collecting common sense information from volunteers on the Internet. This information is converted, using automatic NLP techniques, to a semantic network called ConceptNet. Over the years ConceptNet has grown to contain over 250,000 predicates in English and has recently been expanding to include many new languages.

Using principal component analysis on the graph structure of ConceptNet yields AnalogySpace, a vector space representation of common sense knowledge. This representation reveals large-scale patterns in the data, while smoothing over noise, and predicts new knowledge that the database should contain. The inferred knowledge, which a user survey shows is often correct, is used as part of a feedback loop that shows contributors what the system is learning, and guides them to contribute useful new knowledge.

We feel that information retrieval would benefit from our work in several ways. First, interfaces used in IR could benefit from the "sanity checking" features that adding common sense to a system provides. In the past, this has been used in speech recognition, predictive text entry, and other UI applications. Secondly, we would like to explore a representation similar to AnalogySpace, or even built on it, for other types of complex data such as those found in IR. We feel that AnalogySpace and principal component analysis shows great potential in reasoning which can extend to other areas.

Normalized Clarity and Guided Query Interpretation

Daniel Tunkelang
Endeca

For the past three decades, most research in information retrieval has assumed a ranked retrieval model, in which a query returns a ranking of corpus documents by their estimated relevance to this query. This model maps to the familiar user interface of most commercial and academic search engines.

Despite its popularity, the ranked retrieval model suffers because it does not provide a clear split between relevant and irrelevant documents. This weakness makes it problematic to obtain even basic analysis of the query results, such as the number of relevant documents, let alone a more complicated one, such as the result quality.

In contrast, a set retrieval model partitions the corpus into two subsets of documents: those that are considered relevant, and those that are not. A set retrieval model does not rank the retrieved documents; instead, it establishes a clear split between documents that are in and out of the retrieved set. This explicit determination of the set of documents that constitutes the query results makes it possible to analyze this set, and then to apply this analysis to improve user experience.

In particular, we can revisit an analysis technique designed to measure result quality—namely, the query clarity score introduced by Cronen-Townsend and Croft in 2002 [1]. This score evaluates query results based on the information gain observed in the set of top-ranked query results. In theory, such analysis of query results could be used to determine the success of the system in answering a query. However, research by Turpin and Hersh in 2004 [2] showed a lack of correlation between clarity scores and user performance.

We claim that query clarity is an effective measure, but that it needs to be revised to explicitly leverage a set retrieval model. We propose a normalized clarity score that measures the clarity of a query result set relative to similarly sized document sets.

We present our work in progress on normalized clarity, showing both theoretical and empirical results. We also outline an application of normalized clarity to guide a process of interactive query reformulation.

[1] Cronen-Townsend, S. and W. B. Croft, W. B. Quantifying query ambiguity. In Proc. of Human Language Technology 2002, pages 94--98, March 2002.

[2] Turpin, A. and Hersh, W. Do clarity scores for queries correlate with user performance?. In Proceedings of the 15th Australasian Database Conference - Volume 27 (Dunedin, New Zealand). K. Schewe and H. Williams, Eds. ACM International Conference Proceeding Series, vol. 52. Australian Computer Society, Darlinghurst, Australia, 85-91, 2004.

Less Searching, More Finding: Improving Human Search Productivity

William A. Woods
ITA Software

Finding information and organizing information so that it can be found are two key aspects of any company's knowledge management and knowledge delivery strategy. This talk will describe what I have learned from years of thinking about these problems and from a project I led at Sun Microsystems Laboratories that addressed these problems by combining the respective strengths of humans and computers in a knowledge-based system to help people find information. It explored a new search technology aimed at addressing problems that hinder human search effectiveness, and it developed techniques that provide a user with the necessary information to quickly decide whether a document has the information being sought. Unlike many previous attempts to improve search effectiveness, this system demonstrated a substantial improvement in human search productivity.

The system combines a technique for automatically constructing a semantic conceptual index of the material to be searched with a new passage retrieval algorithm that finds specific passages of text that are likely to contain the information sought. The first technique can supplement or replace expensive manual indexing of material, and it provides a semantically oriented conceptual structure that can be intuitively browsed by information seekers. The information in this conceptual index can also be used by the passage retrieval algorithm for find passages that are relevant to a request but use different, semantically related terms than those used in the query.

The system makes use of linguistic and world knowledge and exploits sophisticated knowledge representation techniques. It combines linguistic knowledge and natural language processing with knowledge representation techniques to automatically construct an intuitive conceptual taxonomy of all the words and phrases found in the material, augmented with additional semantically related terms, and organized by generality. More general terms occur higher in the taxonomy, while more specific terms are linked below more general terms that "subsume" them. This provides useful information for the search algorithm, which automatically includes any terms that are subsumed by the requested query terms, and it is also an intuitive structure for human browsing. The system provides an integrated framework for combining searching and browsing and allows a user to switch easily between the two perspectives.

Why is searching so difficult?

For years I have been asking myself this question, beginning when I took a course in information retrieval as a graduate student and was appalled to discover what information retrieval systems actually did. I expected the system to understand what I was asking about and find documents that were about that. I discovered that all these systems did was count occurrences of words and push the numbers through a simple equation to compute a ranking. Since then, I have been trying to develop systems that come closer to my original assumption. It turns out that this involves a lot of linguistic and cognitive science insights and the use of some real knowledge (in addition to good algorithms and human factors).

One of the problems that makes searching difficult is that people often ask for information using different terminology from that used in the material that they need to find. To be successful a search engine needs to use knowledge to make connections between what the user asks for and what they need to find. Two sources of such connections are morphological and semantic relationships. These are addressed by automatically constructed conceptual taxonomy mentioned above. Another problem is that of finding out whether a retrieved document actually has the information that you are seeking. This is addressed by the specific passage retrieval algorithm and features provided in the human interface to the system. Both techniques will be described, together with some examples and some experimental results.

References:

"Linguistic Knowledge can Improve Information Retrieval," William A. Woods, Lawrence A. Bookman, Ann Houston, Robert J. Kuhns, Paul Martin, and Stephen Green, Proceedings of ANLP-2000, Seattle, WA, May 1-3, 2000, (preliminary version: Technical Report SMLI TR-99-83, Sun Microsystems Laboratories, Mountain View, CA, December, 1999. Online at: http://www.sun.com/research/techrep/1999/abstract-83.html

"Aggressive Morphology for Robust Lexical Coverage," William A. Woods, Proceedings of ANLP-2000, Seattle, WA, May 1-3, 2000, (preliminary version: Technical Report SMLI TR-99-82, Sun Microsystems Laboratories, Mountain View, CA, December, 1999. Online at: http://www.sun.com/research/techrep/1999/abstract-82.html)

"Conceptual Indexing: Practical Large-Scale AI for Efficient Information Access," William A. Woods, Proceedings AAAI 2000, Austin, TX, August 2, 2000.

"Searching vs. Finding," William A. Woods, ACM Queue, Vol. 2, Issue 2 (April 2004), pp 27-35. (available online at: http://portal.acm.org/citation.cfm?id=988405, or at http://acmqueue.com/modules.php?name=Content&pa=showpage&pid=137).

Mediating between User Query and User Model with Adaptive Relevance-Based Visualization

Jae-wook Ahn and Peter Brusilovsky
School of Information Sciences, University of Pittsburgh
{jaa38, peterb}@pitt.edu

Abstract

Personalized information retrieval systems [1] seek to adapt search results to long term interests of an individual users represented in a user profile (also known as user model). One of the problems of these systems is how to "fuse" query-based and profile-based document rankings in search result presentation. The traditional solution to this problem, which is applied in several adaptive search systems, is to select a fixed mediation point α between 0 and 1 and to produce a personalized rank by fusing query- and profile-based rankings with coefficients α and (1-α). By manipulating α, the system designers can give more priority to documents similar to the query or documents similar to the profile. This paper presents a more flexible approach to "fusing" query- and profile-based rankings. The idea of this approach is to allow the analysts to dynamically decide whether they are interested in documents which are closer to the query or documents which are closer to the user profile – with the ability to navigate on a continuum between the query to the user profile and back again.

The core component of our approach is the relevance-based visualization originally implemented in VIBE [2]. VIBE is known as an excellent tool for visual query results analysis. VIBE supports POI (Point of Interest)-based browsing. POIs represent key concepts or keywords and are displayed as user-draggable icons on the screen. The documents are placed according to their similarities to the POIs. Users can drag and move POIs anywhere they want and the locations of the documents are dynamically updated depending on their similarities to the POIs. It allows the user to explore the connection between search results and query terms, for example, enabling the user to pick a subset of results that is more relevant to a specific query term or group of terms.

In our project we applied relevance-based visualization to help the user to mediate between the query terms and terms from the user profile. *Our key idea is to use both query terms and profile terms as POIs.* The application of the user profile makes the relevance-based visualization adaptive. The results of the visualization are different for different users who have submitted the same query and even different over time for the same user, if the interests of the user represented in the user profile evolve. Our poster presents our implementation of VIBE for adaptive relevance-based visualization, stresses several features that are critical for this type of visualization, and shares some evaluation results. This work is a part of our broader agenda on using adaptive visualization to increase the interactivity and expressiveness of personalized information access systems. The rest of the position paper presents the idea of our approach using a practical example.

Figure 1 shows an example of applying VIBE to the query and profile fusion problem. The circles colored in pink and green are POIs representing two different sets of terms: query terms (pink) and profile terms (green). In this example a user entered a query "NUCLEAR WEAPON" and the system retrieved relevant articles with high similarity scores (which will be discussed in detail in Section 3). White squares represent these retrieved documents and users can examine their titles and summaries by hovering the mouse cursor over the square icons. We extracted 10 profile terms and displayed the top 5 of them as green circles on the screen. The rest of the profile terms are disabled temporarily and docked in a white box at the corner of the screen (4 in this case because one term, NUCLEAR, overlapped the query). Users are able to freely move both query and profile terms and explore which document is related to which POI (or term).

This example clearly demonstrates the difference between the traditional search result (query-based ranked list), an adaptively re-ranked result (profile-based ranked list) and our flexible approach exploiting VIBE. Originally, the search engine results contain the top 5 articles on Iranian nuclear weapon development. The ranked list sorts the documents by their relevance score and users typically examine the top ones first. This result is appropriate if the user in this example was most interested in recent events in Iran. However, let's consider a user who is interested in Korean affairs including North Korean nuclear weapon development. Over the weeks of using the system, this user

has been accumulating terms like KOREA, NUCLEAR, JAPAN, and NORTH in her profile of interests. Proponents of adaptive search and filtering systems would argue that this user would be most interested to see information about North Korean, not Iranian nuclear programs and would prefer to see news ranked according to her profile with North Korean news emerging on the top of the list. Unfortunately, in a realistic context it is hard to decide what is the real need of the user because of a lack of information. Her interests may have remained the same (i.e., she does prefer news on North Korean nuclear developments) or may have switched to a different direction (i.e., she is interested in seeing up-to-date news about other programs).

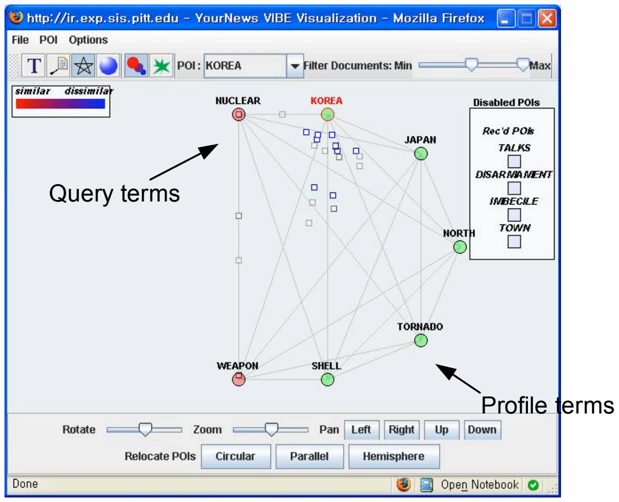


Figure 1. VIBE Visualization for Query and Profile-based Ranking Fusion

"Fusing" query-based ranking and profile-based ranking is a more reliable way to assist the user in an ambiguous context. VIBE allows users interactively explore the query terms, profile terms, and the retrieved documents simultaneously. The users are able to understand the relationships among these three components and discover relevant information more easily. The example above clearly shows the benefits of our approach. By examining the locations of the articles using VIBE, it is surprising that a lot of articles are placed closer to a profile term (KOREA) than the query terms (NUCLEAR and WEAPON). This result is very interesting because the documents visualized here are exactly the same set of articles displayed in the query-based ranked list retrieved by a conventional search engine, where the top 5, most important articles were about Iranian nuclear weapon development. Our approach can provide users with the flexibility to intuitively discern which documents are more related to the query or the profile terms by just glancing at the picture. We don't have to choose either of the two: query or user profile-based ranked list. We can merely show the relatedness of each document to each of the concepts and let users visually explore to understand what the situation really is.

Acknowledgements

This paper is partially supported by DARPA GALE project and by the National Science Foundation under Grant No. 0447083.

References

1. Micarelli, A., Gasparetti, F., Sciarrone, F., Gauch, S.: Personalized search on the World Wide Web. In: Brusilovsky, P., Kobsa, A., Neidl, W. (eds.): The Adaptive Web: Methods and Strategies of Web Personalization. Lecture Notes in Computer Science, Vol. 4321. Springer-Verlag, Berlin Heidelberg New York (2007) 195-230
2. Olsen, K.A., Korfhage, R.R., Sochats, K.M., Spring, M.B., Williams, J.G.: Visualisation of a document collection: The VIBE system. Information Processing and Management 29, 1 (1993)

Characterization of Diagrams and Retrieval Strategies for Them

Robert Futrelle
Northeastern University

There are a few hundred million diagrams available on the web, by rough estimate. They cover every imaginable topic. But quality retrieval of the "diagram you want" is close to impossible, because virtually all current methods rely entirely on the accompanying or referring text to characterize diagram content. Our lab has worked on a variety of aspects of diagrams and their internal content for a number of years, with one of the major goals being how to build IR systems for them. We have published diagram-related papers on machine learning for classification, constraint-grammar-based parsing, ambiguity, summarization, text-diagram interrelations, ontologies, and vectorization of diagram images. Much of our work has been focused on the diagrams that typically appear in papers from the biomedical research literature. This talk will range over the portions of our research most relevant to IR, arguing that many of the topics we have studied need to be kept in mind in building future systems for diagram analysis, representation, interaction, and retrieval.

HUMAN COMPUTATION FOR HCIR EVALUATION

Shiry Ginosar

Endeca Technologies, Inc.
101 Main Street, Cambridge, MA 02142
sginosar@endeca.com

ABSTRACT

A novel method for the evaluation of Interactive IR systems is presented. It is based on Human Computation, the engagement of people in helping computers solve hard problems. The Phetch image-describing game is proposed as a paradigmatic example for the novel method. Research challenges for the new approach are outlined.

***Index Terms*—** Interactive IR and HCIR evaluation, Web-based games.

1. INTRODUCTION

There are currently two main approaches to evaluation of IR systems - the TREC conference approach and the HCI approach, and neither is optimal across the wide range of systems that exist today. In particular, evaluation paradigms for Interactive IR systems are interesting to investigate, since on the one hand the TREC evaluation method cannot be applied here [7,8] while on the other hand HCI methods tend to be hard to generalize.

In this paper, we consider the relative value of the two primary approaches to this problem and propose and discuss a novel approach to evaluating IR and Interactive IR systems that uses Human Computation [1]. This approach extends TREC evaluation metrics so that it can be applicable to interactive systems, and it improves upon HCI methods by reducing their subjectivity.

2. EXISTING IR EVALUATION METHODS

The first approach for IR systems evaluation, taken by TREC [http://trec.nist.gov] is based on a batch evaluation. The queries and corpus to be used are decided upon *a priori* and the entire corpus is relevance-ranked by hand for each of the queries. Each IR system is then queried using a batch process with the pre-compiled queries over the given corpus. The resulting relevance-ranked set of documents is then compared to the pre-annotated "gold standard" and scores such as *precision* and *recall* are computed [10,13]. The batch process approach is arguably a successful measure of goodness for the effectiveness of the IR system itself [10,11].

However, evaluating an IR system using a batch process may fail to capture the intended use of systems that are designed to support other information discovery processes [13]. This is especially true in regards to evaluating Interactive IR systems. On the one hand, classic IR evaluation relies on a one click paradigm where queries are first composed in full and then sent to the systems to compute a static set of answers [10,13]. On the other hand, Interactive IR systems are often designed to enable a user to iteratively formalize the query. Since the query as a whole is not known *a priori*, there is no way to assess the relevance of documents in the corpus in advance and therefore there is no way to compose a gold standard with which to compare results returned from different systems. Thus, alternative methods must be used in order to evaluate such systems [7, 8].

The second approach to evaluation of IR systems, used primarily within the HCI community, focuses on task level evaluations rather than evaluating the results for individual queries. Such evaluations often employ a mix of objective and subjective metrics such as completion time, user satisfaction and perceived user success [8,14]. Since the metrics used by HCI are partially subjective and since the tasks performed during the evaluation are highly correlated with the specific system and the specific corpus used [8,14], it is hard to compare different systems and the results of these evaluations are rarely accepted by the greater IR community.

Furthermore, HCI evaluations that are set up as user studies are often stymied by the lack of willing participants, the need to compensate participants and the difficulties of recruiting participants from outside the specific university or company where the study is conducted. These hardships can result in lack of data or lack of a sufficiently varied participant population, both of which make it difficult to make statistically significant claims.

3. HUMAN COMPUTATION EVALUATION OF IR

In this paper, we propose a new approach to evaluation of IR and Interactive IR systems. The goal of this line of thought is to design a system that will allow users to perform search tasks in a natural way, while assessing the quality of the system as well as the success and satisfaction of the users in the background. This approach is not intended to achieve a mapping to the classical evaluation scores used by TREC (unlike [11] which claim to successfully do so, or [7,13] which claim that there is no correlation between user success and TREC metrics). Rather, we seek a new scoring system that will be able to compare different user-systems combinations.

Moreover, to overcome the hardships of recruiting individuals for participation in user studies, we propose to incorporate the concept of Human Computation [1] into the design of our system. Human Computation engages people to aid computers in completing tasks which are either too hard or too expensive for computers to do on their own. Most Human Computation systems are designed as games [1,2,3,4,5,6] which people enjoy playing, or as verification systems which act as gateways to information that people want to access [http://www.captcha.net/, http://recaptcha.net/]. However, a Human Computation system is more than a game: it is cleverly designed such that as a side effect of game play or everyday tasks such as logging in to an email account, useful information can be collected.

4. AN EXAMPLE EVALUATION USING A GAME

As an example of the Human Computation evaluation paradigm, we will investigate in more detail the possible use of the online game Phetch [4] that can be hooked up to different IR systems [3]. Phetch requires players to perform search tasks in order to advance in the game. In Phetch, a describer generates a text description of an image and multiple seekers race to identify the described image out of a large collection of similar images. People play the game because it is fun, and as a side effect of game play the set of IR systems supporting the game may be evaluated. Since the game is interactive in nature, this type of evaluation is suited for IR as well as Interactive IR systems.

There are several advantages for using a game like Phetch for evaluation. First and foremost, since the game involves users performing search tasks while trying to fulfill an information need, it naturally lends itself to evaluation of not only IR systems but also of Interactive IR systems.

Second, the search task itself within the game is done in a natural way. Players are presented with an item (in this case, an image) that they need to find, and they are expected to devise their own ways in which to find it. This type of search task is very similar to search tasks that users of IR systems perform in real life scenarios and therefore would eliminate the need to come up with a contrived simulated work task situation for the purpose of the evaluation [9].

Third, a game like Phetch outputs a clean scoring number for players in the game. This score encapsulates the success of the player both as a seeker who searches for images as well as a describer who describes images for others to find. It depends on the randomly chosen image as the goal of the search, on the speed in which the player processes visual and language information, on the opponents she played against and even on the speed of her internet connection. However, an average scoring over many players, many images and many game sessions could potentially serve as a form of measure of goodness for the combination of a generic user with the specific IR system that was hooked onto the game. This scoring could later be incorporated with other metrics from HCI user studies or batch processes performed against all or part of the IR system to produce a more accurate metric. Repeating the same setting of game play with the same corpus using other IR systems would produce similar scoring which could then be compared with the first, resulting in an overall comparison between the two IR systems and the ways in which they allow users to interact with them.

In this way, a Human Computation system could potentially bridge between the two different approaches to IR evaluation. It could provide a clean score to aid the current ways of comparing different IR systems while also taking into account user interaction with the system as well as the performance of the system itself.

From our experience with Phetch we learned that it can be employed as a possible Human Computation evaluation tool, but an interesting problem is how to apply the concepts from Phetch to non-image domains, in particular text documents.

5. CHALLENGES AND OPEN QUESTIONS

Using Human Computation for evaluation of IR systems requires further research. In particular, this paradigm should be correlated with accepted figures of merit of IR systems that are used by TREC and HCI methods, such as accuracy, precision, recall, success and satisfaction of users.
Additional work may be required for interfacing a Human Computation game with other types of IR systems. For example, in systems that support faceted metadata browsing such as Flamenco [14] and Endeca's [www.endeca.com] Guided Navigation, the corpus should be pre-processed to organize flat tags hierarchically, for which many automatic and semi-automatic methods are available [12]. It is clear that when applying different IR interfaces to the same corpus, the quality of data preprocessing to tailor it to the specific interface could dramatically impact the results of the evaluation. Dealing with preprocessing could potentially be achieved in a manner similar to the one taken by TREC, where competing teams are required to submit a system that interfaces with the Game. The corpus would be known ahead of time, so each team could do their best effort on data preprocessing. Assuming no *a priori* knowledge of the

corpus may also lead to interesting results, but is not currently under consideration.

6. ACKNOWLEDGEMENTS

I thank Daniel Tunkelang, Michael Tucker, Robin Stewart and Andrew Schlaikjer for their insightful comments. I further thank Dr. Luis von Ahn for introducing me to the exciting domain of Human Computation.

7. REFERENCES

[1] von Ahn**,** L. Games With A Purpose. In *IEEE Computer Magazine,* June 2006*,* pp. 96-98.

[2] von Ahn, L., and Dabbish, L. Labeling Images with a Computer Game. In *ACM Conference on Human Factors in Computing Systems (CHI),* 2004, pp. 319-326.

[3] von Ahn, L., Ginosar, S., Kedia, M., and Blum, M. Improving Image Search with Phetch. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* 2007, Vol. 4 pp. IV-1209-IV-1212.

[4] von Ahn, L., Ginosar, S., Kedia, M., Liu, R., and Blum, M. Improving Accessibility of the Web with a Computer Game. In *ACM Conference on Human Factors in Computing Systems (CHI),* 2006, pp. 79-82.

[5] von Ahn, L., Kedia, M., Liu, R., and Blum, M. Verbosity: a game for collecting common-sense facts. In *ACM Conference on Human Factors in Computing Systems (CHI),* 2006, pp. 75 – 78.

[6] von Ahn, L., Liu, R., and Blum, M. Peekaboom: a game for locating objects in images. In *ACM Conference on Human Factors in Computing Systems (CHI),* 2006, pp. 55-64.

[7] Al-Maskari., A. Beyond Classical Measures: How to Evaluate the Effectiveness of Interactive Information Retrieval System?. In *ACM Special Interest Group on Information Retrieval (SIGIR),* 2007, pp. 915.

[8] Borlund, P., The IIR Evaluation Model: A Framework for Evaluation of Interactive Information Retrieval Systems. *Information Research,* 2003, Vol. 8, No. 3.

[9] Borlund, P., and Ingwersen, P. The Development of a Method for the Evaluation of Interactive Information Retrieval Systems. *Journal of Documentation,* 53(3), 1997, pp. 225-250.

[10] Cleverdon, C. The Cranfield Tests on Index Language Devices. *Aslib Proceedings,* 19:173-192, 1967. (Reprinted in K. Spark Jones and P. Willet, editors. *Readings in Information Retrieval.* Morgan Kaufmann Publishers Inc., 1997)

[11] Huffman, S., and Hochster, M. How Well does Result Relevance Predict Session Satisfaction?. *ACM Special Interest Group on Information Retrieval (SIGIR),* 2007. pp. 567-573.

[12] Stoica, E. and Hearst, M. Nearly Automated Metadata Hierarchy Creation. *HLT-NAACL,* 2004. Companion Volume.

[13] Turpin, A., and Scholer, F. User Performance versus Precision Measures for Simple Search Tasks. *ACM Special Interest Group on Information Retrieval (SIGIR),* 2006. pp. 11-18.

[14] Yee, K.P., Swearingen, K., Li, K. and Hearst, M. Faceted Metadata for Image Search and Browsing. In *ACM Conference on Human Factors in Computing Systems (CHI),* 2003.

Jigsaw: a Visual Index on Large Document Collections

Carsten Görg John Stasko
School of Interactive Computing & GVU Center
Georgia Institute of Technology
{goerg,stasko}@cc.gatech.edu

Abstract

Investigative analysts who work with collections of text documents connect embedded threads of evidence in order to formulate hypotheses about plans and activities of potential interest. As the number of documents and the corresponding number of concepts and entities within the documents grow larger, sense-making processes become more and more difficult for the analysts. We have developed a visual analytic system called *Jigsaw* that represents documents and their entities visually in order to help analysts examine reports more efficiently and develop theories about potential actions more quickly.

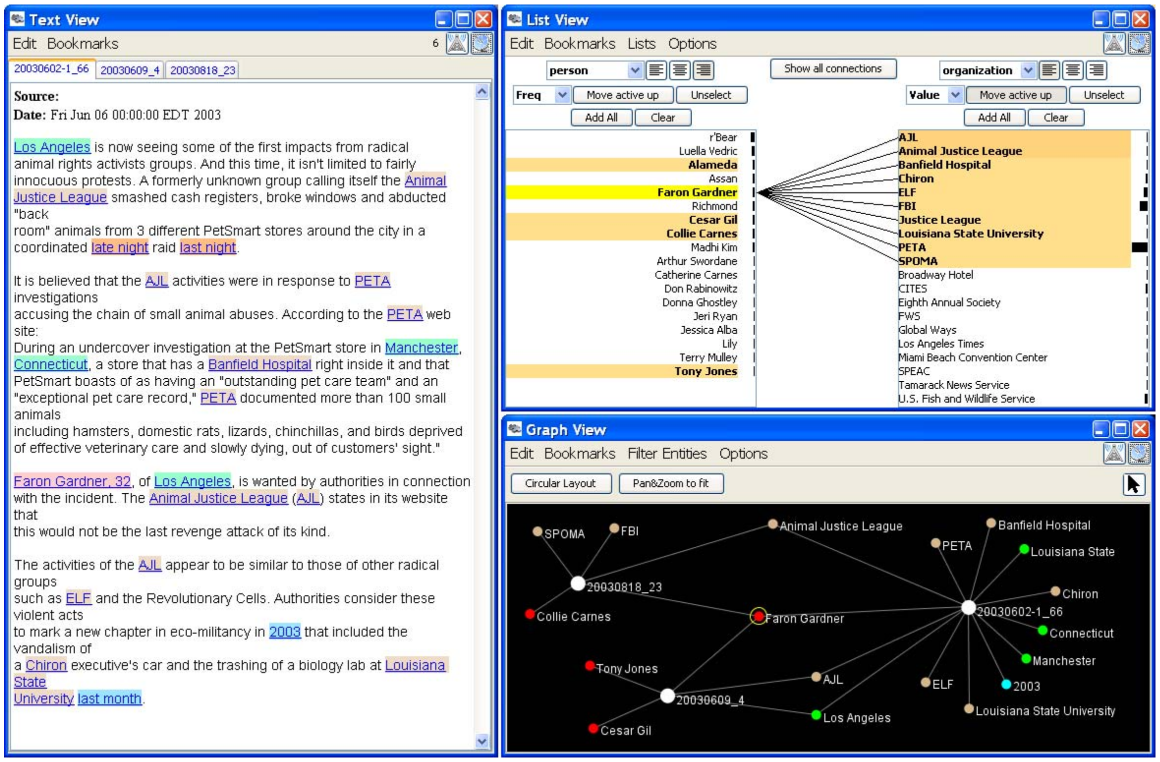
The *Jigsaw* system provides multiple coordinated views that show connections between entities (like people, places, organizations, dates, *etc.*) across documents. A connection between two entities is defined as a co-occurrence in at least one document.

To allow *Jigsaw* to handle large datasets, the system does not show the entire dataset at once but uses an incremental query-based approach to show a subset of the dataset. The query window allows analysts to search for entities and also provides a text search within the documents. *Jigsaw*'s query approach is different from traditional search engines. Getting a list of ranked documents as a result of a query would not be sufficient for analysts' tasks because their activities go beyond just looking for a set of documents. Analysts also care about understanding what is inside of a document and how those entities are connected to entities in other documents. To support that task *Jigsaw* acts as a visual index on the document collection: the query results, consisting of entities and documents, are sent to multiple views that show different perspectives on the connections between those elements. The analysts can interact with the views, apply filters, or expand the context to gain more insight about the document collection. This exploration then spurs further queries and retrieves other documents and entities. While acting as a visual index, *Jigsaw* guides the analysts to related documents and facilitates the information retrieval process.

Jigsaw presents documents and entities resulting from queries through six different types of views. Therefore, the availability of significant screen space is very beneficial. The Text View shows document text, allowing analysts to validate connections, providing their context, and giving access to information that is not extracted as an entity. The List and Graph Views display connections between entities and allow analysts to explore the connection network. The Scatter Plot View highlights pairwise relationships between any

two entity types. The Time Line and Calendar Views organize entities and reports by date to ease the search for time patterns.

Figure 1 shows three different views after querying for “Faron Gardner” and exploring the query result. The Text View shows documents related to the query in three tabs. Entities within those documents are color coded accordingly to their type. The List View shows people and organizations connected to Gardner, with a darker shade of orange indicating a stronger connection. The Graph View displays the documents in which Gardner is mentioned, as well as the entities within these documents. Thus, it is easy to see which entities are mentioned in multiple documents.



Source:
Date: Fri Jun 06 00:00:00 EDT 2003

Los Angeles is now seeing some of the first impacts from radical animal rights activists groups. And this time, it isn't limited to fairly innocuous protests. A formerly unknown group calling itself the Animal Justice League smashed cash registers, broke windows and abducted "back
room" animals from 3 different PetSmart stores around the city in a coordinated late night raid last night.

It is believed that the AJL activities were in response to PETA investigations
accusing the chain of small animal abuses. According to the PETA web site:
During an undercover investigation at the PetSmart store in Manchester, Connecticut, a store that has a Banfield Hospital right inside it and that PetSmart boasts of as having an "outstanding pet care team" and an "exceptional pet care record," PETA documented more than 100 small animals
including hamsters, domestic rats, lizards, chinchillas, and birds deprived of effective veterinary care and slowly dying, out of customers' sight."

Faron Gardner, 32, of Los Angeles, is wanted by authorities in connection with the incident. The Animal Justice League (AJL) states in its website that
this would not be the last revenge attack of its kind.

The activities of the AJL appear to be similar to those of other radical groups
such as ELF and the Revolutionary Cells. Authorities consider these violent acts
to mark a new chapter in eco-militancy in 2003 that included the vandalism of
a Chiron executive's car and the trashing of a biology lab at Louisiana State University last month.

Figure 1: The Text View, List View, and Graph View showing different perspectives after querying for “Faron Gardner” and exploring the query result.

Jigsaw’s views are coordinated using an event mechanism: interactions with one view (selecting, adding, removing, or expanding entities) are transformed into events that are then broadcast to all other views. Thus, the views of the system stay consistent and provide different perspectives on the same data.

For a detailed description of the system, we refer the reader to an article[1] about *Jigsaw* in the VAST '07 proceedings and to a video on the project website[2] that shows interaction with the system.

---

[1] J. Stasko, C. Görg, Z. Liu, and K. Singhal. Supporting Investigative Analysis through Interactive Visualization. In *IEEE Symposium on Visual Analytics Science and Technology*, October 2007.
[2] Jigsaw project. http://www.gvu.gatech.edu/ii/jigsaw/.

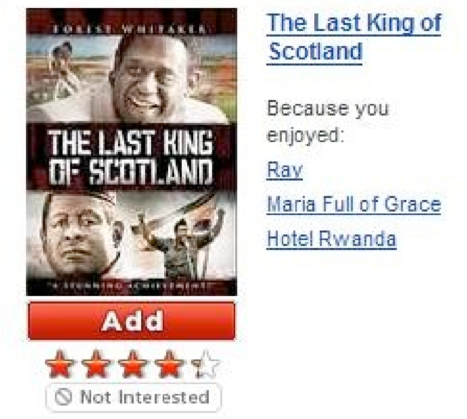
Reasoning and Learning in Mixed-Initiative Tasks

Yifen Huang (hyifen@cs.cmu.edu)
Carnegie Mellon University

Information management becomes more and more challenging with information communicated electronically or retrieved from the web. A human user is often overwhelmed by the amount of information that is easily accessible to them. The approach of Human-Computer Interaction (HCI) designs interfaces to leverage off a user's efforts. On the other hand, Artificial Intelligence (AI) develops more autonomous techniques in information retrieval, data mining, and learning. Although both approaches aim toward the same goal, they have proceeded in the past with too little interaction.

In a simplified notion, a computer has two capabilities: (1) an inference function that takes inputs, e.g., a set of documents, and produces outputs, e.g., a rank list of these documents, and (2) representation of inference results. The first capability is AI related and the second capability is HCI related. I want to propose a new mixed-initiative framework that involves both approaches and hopefully boosts the quality of the solution jointly. The idea is as the following: in addition to a computer, a user can also give feedback on what representation we show to the user and the inference function can be re-trained based on user feedback. This forms a directed loop from the AI approach to the HCI approach to the user and back to the AI approach.

A following question is what kind of information can be passed through this loop and how it contributes to the overall performance. We have found it is quite useful to extract properties of the inference function and show these hypothesized properties to a user because a user is often clueless if only presented by inference results.



For example, Netflix recommended a movie, The last king of Scotland, to me. Since I was not familiar with the recommended movie, I had no idea if it was a good recommendation or not if this were the only information I got. However, Netflix showed me that it suggested this movie because of some movies I ranked high before. Despite the fact I hadn't heart about this recommended movie, I got the basic idea before clicking into its introductory page because I found similarities between previous movies. Showing previously enjoyed movies is an example of representing hypothesized properties of inference functions.

Furthermore, a system can collect user feedback on not only inference results but also hypothesized properties of inference functions. User feedback can be interpreted as modified properties. With proper model adaptation, we can re-train an inference function so its new hypothesized properties gets closer to a user's modified properties on the same task. For example, if Netflix provides a way for me to say "Maria Full of Grace and Hotel Rwanda are an interesting movie pair for this recommended movie," we can use this modified property in future recommendations -- when the system finds two new recommendations: movie A is associated with Maria Full of Grace and Hotel Rwanda, and movie B is associated with Maria Full of Grace and Ray. The system can learn by past feedback that movie A is probably a better recommendation because it is associated with a better property.

Last, we would like to keep previously obtained user feedback in the loop so a user has no need to give the same feedback twice. This can be done by validating if previous modified properties still exist in current inference results or hypothesized properties. If they still exist, we can show them in the interface to remind a user about their previous decisions.

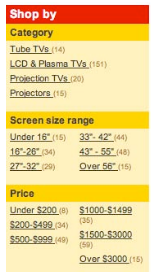
Resonance: Introducing the concept of penalty-free deep look ahead with dynamic summarization of arbitrary result sets

Blade Kotelly
Endeca Technologies
101 Main St. Cambridge, MA, USA
blade@endeca.com

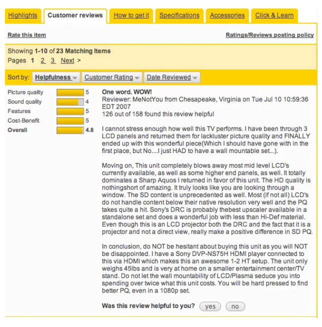
INTRODUCTION

Providing the ability for someone to be able to see what's around the corner is a hallmark of good HCIR. A great example of this is when there are various refinements that a user could make to refine a set, and next to each refinement is a count which tells the user how many results would be left in the set if they choose that particular refinement.

EXAMPLE:



The refinement count next to "Projection TVs" tells me that if I chose that option, my next set would contain only 20 items. This kind of look-ahead is very useful and easy to understand. But how can I give a similar ability for a user to see what's under the covers when it comes to learning about what people are saying about a product? Right now, there are a plethora of sites that use user-reviews and user-ratings to help people evaluate a product.



Showing 1-10 of 23 Matching Items
Pages 1 2 3 Next >

One word. WOW!
Reviewer: MeNotYou from Chesapeake, Virginia on Tue Jul 10 10:59:36 EDT 2007
126 out of 158 found this review helpful

I cannot stress enough how well this TV performs. I have been through 3 LCD panels and returned them for lackluster picture quality and FINALLY ended up with this wonderful piece(Which I should have gone with in the first place, but No....I just HAD to have a wall mountable set...).

Moving on, This unit completely blows away most mid level LCD's currently available, as well as some higher end panels, as well. It totally dominates a Sharp Aquos I returned in favor of this unit. The HD quality is nothingshort of amazing. It truly looks like you are looking through a window. The SD content is unprecedented as well. Most (if not all) LCD's do not handle content below their native resolution very well and the PQ takes quite a hit. Sony's DRC is probably thebest upscaler available in a standalone set and does a wonderful job with less than Hi-Def material. Even though this is an LCD projector both the DRC and the fact that it is a projector and not a direct view, really make a positive difference in SD PQ.

In conclusion, do NOT be hesitant about buying this unit as you will NOT be disappointed. I have a Sony DVP-NS75H HDMI player connected to this via HDMI which makes this an awesome 1-2 HT setup. The unit only weighs 45lbs and is very at home on a smaller entertainment center/TV stand. Do not let the wall mountability of LCD/Plasma seduce you into spending over twice what this unit costs. You will be hard pressed to find better PQ, even in a 1080p set.

Was this review helpful to you?

These reviews are very helpful, but the problem starts when a user is confronted with a number of reviews that is too numerous to read in a timely manor. Often there are several reviews, from as few as 20 to more than 70, associated with a single product on sites like CircuitCity.com. If you were to look at a set of televisions say, all 151 plasma TVs, you'd see 363 reviews covering the first 10 products alone! Even if you get down to one product, but see that the product has more than 100 reviews it wouldn't be reasonable to expect most users to read them all. They'll typically read the most favorable reviews (why should I buy it?) and the least favorable ones (what's wrong with it?) If you can winnow the set of reviews down by some method, perhaps by looking only at reviews from people who consider themselves professional users, or people who are in a particular age range, then the remaining reviews, which would normally be unread, would become more useful. The problem is that you can't examine the reviews before you start reading by any method that will give you a sense of what the reviews are talking about, in general, and you can't read the reviews across multiple products to see what people are talking about within a certain category to compare 2 arbitrary categories against each other (e.g., what are people saying about large, Panasonic plasma TVs compared to large Sony TVs? Or what are people saying about large, Panasonic TVs compared to small, Panasonic TVs?)

PROPOSED SOLUTION

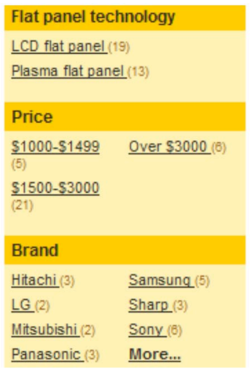
Here is a proposed technique that could be used to provide some deep look ahead that would give the user an understanding of what people are saying, in aggregate, about an arbitrary collection of items.

Applying certain technologies, it's easy to automatically go through an unstructured document and extract terminology, such as a set of terms associated with reviews of consumer products, like "banding" and "exceptional picture quality". It is also possible to cluster those phrases so that they form logical groupings. For example, when clustering technology is used to sort through all the text returned from a query on Wikipedia for the term Apple, we see that term-based clusters are formed in which one cluster has words like, core, juice production, etc. and a different cluster has words like, iPod, Computer and Steve Jobs.

If we can take the same technology and apply it to the unstructured text of a set of user reviews, we could again extract all the terms in the set of reviews, form clusters, and present those to the user so that they could use a cluster to further filter the set of reviews. However, this isn't that useful because it would require the user to attempt to understand the meaning about each cluster and then determine if choosing that cluster would help them get to a set of reviews that were talking about a particular topic. Also, when terms and phrases are presented without context, the user doesn't know the valence of a particular idea, i.e., "**contrast ratio**" could be used in the phrase "phenomenal **contrast ratio**" as well as "the worst **contrast ratio** of all the TVs on the market."

A more useful interaction would be for the system to automatically extract the terms, and perhaps when the user rolls over a dimension refinement (e.g., " 33" – 42" (8)) they would be presented with a short set of snippets which talked about the various hot topics in that set. It's almost as if we were able to get people in a room, with some arbitrary set of televisions (in this case, say the 8 available televisions which are between 33" and 42" inches) and see what they were talking about the most – to understand what thoughts would resonate in the head of a person listening to the reviewers in that room.

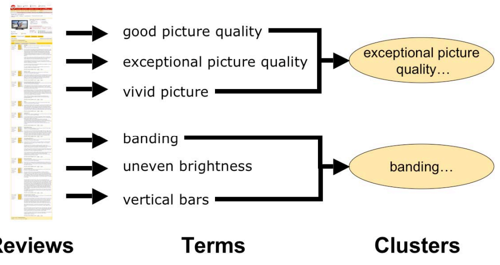
This solution would primarily be used when the domain is somewhat specific (televisions, compared to all of consumer-electronics) and where a user would normally get stuck as to what they can do to advance their ability to dive deeper. When I need to choose a television, I can easily get myself into a position where I'm not sure what to do next. In the below example, the split between LCD and Plasma TVs is almost even, the price range is heavily centered around the $1500-3000 zone, ad the brands all have just a few televisions to choose from.



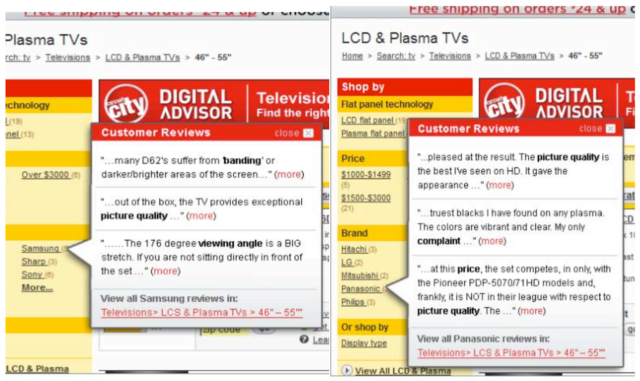
So if I don't care if I get an LCD or Plasma TV, if I know that it will cost me between $1500 and $3000 and I am not brand loyal, then I have to start looking at all my results for other things that will help me further refine my set. It's at this point that I could really use a way to look more deeply at the results, and take advantage of all the unstructured information from things like user-reviews or other collections of information.

To accomplish this, we could take all the terms from all the reviews in the result set (say, 8 televisions, which have a total of 250 reviews among them), cluster the all the terms from all the reviews and then apply the terms found in each cluster as a set of search-terms against the whole corpus of 250 reviews from those 8 products. We'd then rank the results (presumably with the top result as the one which is most about the terms in a particular cluster) and present the user with a snippet that contains the searched-terms from the top ranked review.

Using the reviews to generate terms, then clusters from the terms:



We could then present the snippets in a readable form, which also indicated to the user the set from which the snippets came by presenting them when the user rolled over a dimension value:



Customer Reviews

"…many D62's suffer from '**banding**' or darker/brighter areas of the screen…" (more)

"…out of the box, the TV provides exceptional **picture quality** …" (more)

"……The 176 degree **viewing angle** is a BIG stretch. If you are not sitting directly in front of the set …" (more)

View all Samsung reviews in:
Televisions> LCS & Plasma TVs > 46" – 55""

Customer Reviews

"…pleased at the result. The **picture quality** is the best I've seen on HD. It gave the appearance …" (more)

"…truest blacks I have found on any plasma. The colors are vibrant and clear. My only **complaint** …" (more)

"…at this **price**, the set competes, in only, with the Pioneer PDP-5070/71HD models and, frankly, it is NOT in their league with respect to **picture quality**. The …" (more)

View all Panasonic reviews in:
Televisions> LCS & Plasma TVs > 46" – 55""

This allows people to be able to make judgments about what the information in the reviews indicates about an arbitrary set and compare that to any other set (or example comparing 2 different manufacturers) without having to click through and read all – or any of the reviews. Even better, is that the user won't see the three or four most common phrases, which in a set of televisions might be very similar.

However, problems with this technique might occur when people don't have the ability to directly compare the snippets against each other because the clusters are inherently about different things, i.e., while one set of snippets of reviews may talk about *price* and *picture quality*, another set may focus on *ease-of-set-up* and *viewing angle*. That difference could lead a user to think that one TV is a better value, while the other is easier-to-use, which may in fact, not be accurate.

Visual Concept Explorer

Xia Lin
College of Information Science and Technology
Drexel University
Philadelphia, PA 19104

Visual Concept Explorer (VCE) is a visualization system developed to explore potentials of visual mapping and information retrieval. Implemented as a client-server application, VCE contains a Java-based server that interacts with a very large ontology, the National Library of Medicine's UMLS Knowledge Source (UMLS, 2007) and a live search engine, the free PUBMED search engine available on the Web. Its visual interface is implemented in FLASH with various mapping and interactive functions. Figure 1 is a sample interface screen for searching the keyword "cognition." VCE is available for testing at: http://cluster.cis.drexel.edu/vce/

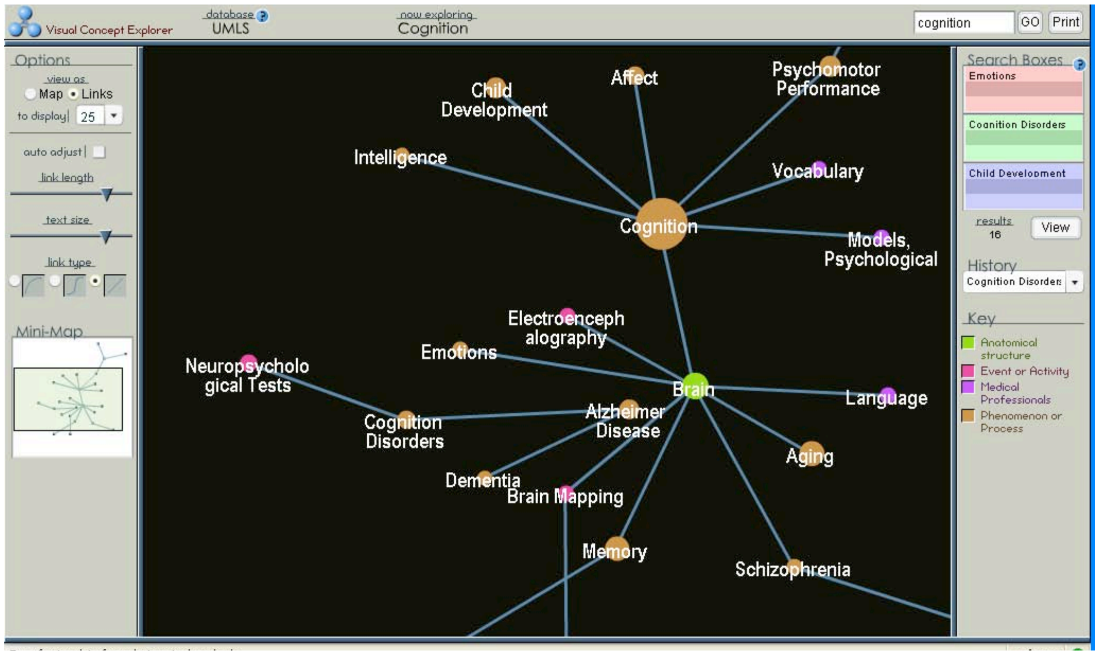


Figure 1. A screen dump of the VCE interface.

Visual Concept Mapping

The first function of VCE is to create dynamic visual concept maps for user's queries. When a user's query term is received, VCE first checks the UMLS ontology to see if the term is a standardized term that can be mapped to a unique concept ID. If it is not, the term will be sent to the PUBMED search engine to retrieve documents. The top 20 controlled vocabulary terms (MESH terms) from the top 200 retrieved documents will be presented to the user. The user then can select a term that best matches his or her query for concept mapping. In the example in Figure 1, the user's query term is "cognition" which is a concept term in UMLS. VCE first identifies 24 terms that co-occur most often with the term “cognition” and then map the 25 terms based on some mapping algorithms. The display shown in figure 1 is generated by Pathfinder Associative Networks (PFNET) (Schvaneveldt, 1990). The map clearly shows that the concept “cognition” is most closely related to “brain,” “intelligence,” “vocabulary,” “Models, psychological”, etc., and the term “brain” is connected to “language,” “emotion,” and “electroence phalography,” etc. Such a concept map helps the user explore semantic relationships of terms and navigate through the concept relationships to find the best term to represent his or her information needs. The map is interactive. The user can explore any of the terms shown on the map by double-clicking on it. The clicked term will become the new “focus center” and a new concept map will be generated for the term. The user can also select different mapping models to see concept maps in different styles. Another mapping model currently implemented is the Kohonen’s self-organizing feature map algorithm (Kohonen 1997). Other mapping algorithms can be easily added to the system as needed.

Visual Search Interface
VCE is also a search interface for the PUBMED search engine. The user can use the visual display to develop their search strategies while exploring the term relationships shown on the display. Right-clicking on any of the terms would allow the user to add the term to search boxes on the right-hand side of the screen. Each time a term is added to the search boxes, a search will run automatically against the PUBMED search engine (or other search engines that the user chooses) and the number of hits will be shown or updated on the screen. The search is executed using a Boolean query constructed from the terms in the search boxes – terms in the same box are ORed and terms in different boxes are ANDed together. In this way, the search process becomes a natural result of the user’s interaction and exploration of concept semantics. VCE assists the user in at least three aspects of searching: (1) converting free-text query terms to controlled vocabulary terms, (2) constructing Boolean queries through choosing-and-clicking from dynamic concept maps, and (3) providing immediate feedback of query results. In the example shown, the user starts with the term “cognition” and constructs a query “Emotions AND Cognition Disorder AND Child Development,” yielding 16 results. The query can represent user’s information needs much better.

The VCE prototype will be demonstrated during the conference and viewers are invited to explore various mapping and interactive functions implemented in VCE.

References:

Kohonen, T. (1997). Self-Organizing Maps. Springer-Verlag, Berlin.
Schvaneveldt, R. W. (Ed.). (1990). Pathfinder associative networks: Studies in knowledge organization. Norwood, NJ: Ablex.
UMLS (2007). Unified Medical Language System. http://www.nlm.nih.gov/research/umls/

Work in Progress: Navigating Document Networks

Mark D. Smucker

Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts Amherst
smucker@cs.umass.edu

ABSTRACT

While much research effort has been expended on innovative user interfaces for information retrieval (IR), deployed IR user interfaces have adopted few innovations. Rather than design another novel user interface tool that users never adopt, we decided that our first step would be to better understand the nature of an adopted tool. In that vein, we are in the process of studying the potential and the actual performance of find-similar, which is a widely adopted tool that allows a user to request documents similar to a given document. Find-similar is a compelling IR interface tool for the very reason that users appear to have adopted it and that it has the potential to provide to users the power long known to be available via relevance feedback. Our hope is that by better understanding find-similar, we'll be able to take that understanding and apply it to other user interface tools that will both be powerful and be adopted by users.

1. INTRODUCTION

Find-similar allows a user to request a list of documents similar to a given document. As such, find-similar provides a way for users to navigate from one document to another and browse by document similarity. This feature is typically instantiated as a button or link next to each result in the list of search results. For example, the Excite search engine labeled its find-similar link "More Like This: Click here for a list of documents like this one."

Find-similar can be an important and valuable tool for improving IR systems. Spink et al. [6, 7] analyzed samples of Excite's query logs and reported that between 5 and 9.7 percent of the queries came from the use of the "more like this" find-similar feature. Lin et al. [2] have reported that for the US National Library of Medicine's search engine, PubMed, 18.5% of non-trivial search sessions involve clicks on articles suggested by PubMed's find-similar, which PubMed refers to as *related articles* [3].

While relevance feedback is well known to be a powerful technique for improving retrieval performance, it has seen little adoption by popular search systems. We've shown that find-similar has the potential to match the performance of relevance feedback [4]. Earlier work by Wilbur and Coffee [8] found that certain browsing patterns could improve over the original query's ranking.

HCIR'07 Workshop on Human-Computer Interaction and Information Retrieval, October 23, 2007, Cambridge, Massachusetts.

2. DOCUMENT NETWORKS

Find-similar can be studied and understood in graph theoretic terms. Each document or web page is a node in a graph. When find-similar is applied to a document, find-similar provides the user with links to the similar documents and these links are effectively added to the document. For a web page, these automatically created links join the already existing links on the page. If the added links are good, users should be able to use the links to navigate to other relevant documents. These links make documents in the graph closer to each other, which is good, but these links also increase the amount of time that a user needs to spend examining the page, which is bad.

In a broad sense, find-similar aims to add links to documents such that the time for a user to get from relevant document to relevant document is minimized. These added links can represent many different types of similarity. The most studied form of similarity is content-based, which typically involves comparison of the terms in each document. The web's hyperlinks are another form of similarity; a similarity defined by the authors of the web pages. Find-similar could add links to documents from a similar time period or documents written by the same author, for example.

We've proposed a pair of metrics that can be used to measure the navigability of documents [5] and preliminary experiments show that existing web hyperlinks are ill-suited for navigation from relevant document to relevant document compared to links produced by a content similarity measure.

Different types of content similarity measures create different document networks. Using simulated browsing behaviors, we have found that a query-biased document-to-document similarity consistently outperforms a baseline "regular" similarity. Figure 1 shows an example of how query-biased similarity can result in a better clustering of relevant documents. We intend to study query-biased similarity using our navigability metrics and obtain a better understanding of the more navigable networks produced by query-biased similarity.

3. USER BEHAVIOR

While understanding of the find-similar document networks is important to understanding find-similar and maximizing its performance, we also want to better understand how people use find-similar.

In a study by Huggett and Lanir [1], users found more relevant documents using an interface that provided find-similar over an interface without find-similar. Huggett and Lanir's study used small newswire collections of 2000 doc-

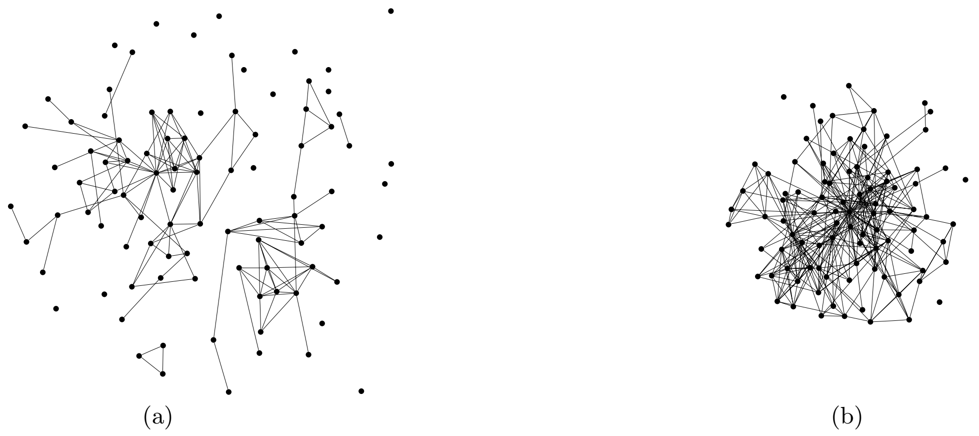


Figure 1: **Simplified depictions of the relevant document networks for TREC topic 337, "viral hepatitis." The network on the left (a) uses regular similarity while the network on the right (b) uses query-biased similarity, which better clusters relevant documents. The documents are closer in figure (b) because they are higher ranked in each other's ranked lists. Links are drawn between two documents when one of the pair is close to the other. The actual relevant document network is a weighted, directed graph [5].**

uments and limited test subjects to two minutes for each search. We would like to examine find-similar's usage on much larger TREC collections and on the web. As Huggett and Lanir did, we will also compare our find-similar implementations to IR systems allowing query reformulation and one of our measures of performance will be the rate at which relevant documents are found. We hypothesize that users adopt IR interface features that provide better rates of information discovery as opposed to tools that may improve ranking performance but overall slow the rate of finding relevant documents.

We also want to learn about how users navigate the document networks formed by find-similar and what forms of user interface support are needed to maximize performance. For example, how far away from the original query will users navigate? Do users apply find-similar to documents that are non-relevant but which they think might lead them to relevant documents? How is find-similar usage interleaved with query reformulation? We intend to answer these and other questions as part of a planned user study.

4. CONCLUSION

Find-similar provides a chance for us to study a user interface feature that has been adopted by search engines and shown to be frequently used by users. To date, we've shown that find-similar has the potential to match a traditionally styled multiple item relevance and that different forms of similarity offer different levels of inherent navigability. Our next steps include a closer examination of similarity functions such as query-biased similarity and actual user studies. As much as possible, we hope to learn what has allowed find-similar to become a useful tool for search when so many other user interface features have failed to succeed and be adopted outside of the laboratory.

5. ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval and in part by the Defense Advanced Research Projects Agency (DARPA) under contract number HR0011-06-C-0023. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor.

6. REFERENCES

[1] M. Huggett and J. Lanir. Static reformulation: a user study of static hypertext for query-based reformulation. In *JCDL '07: Proceedings of the 2007 conference on Digital libraries*, pages 319–328. ACM Press, 2007.

[2] J. Lin, M. DiCuccio, V. Grigoryan, and W. J. Wilbur. Exploring the effectiveness of related article search in pubmed. Technical Report LAMP-TR-145/CS-TR-4877/UMIACS-TR-2007-36/HCIL-2007-10, College of Information Studies, University of Maryland, College Park, July 2007.

[3] Pubmed, `www.pubmed.gov`. "Related articles": `www.nlm.nih.gov/bsd/pubmed_tutorial/m5002.html`.

[4] M. D. Smucker and J. Allan. Find-similar: Similarity browsing as a search tool. In *SIGIR '06*, pages 461–468. ACM Press, 2006.

[5] M. D. Smucker and J. Allan. Measuring the navigability of document networks. In *SIGIR '07 Web Information-Seeking and Interaction Workshop*, 2007.

[6] A. Spink, B. J. Jansen, and H. C. Ozmultu. Use of query reformulation and relevance feedback by excite users. *Internet Research: Electronic Networking Applications and Policy*, 10(4):317–328, 2000.

[7] A. Spink, D. Wolfram, B. J. Jansen, and T. Saracevic. Searching the web: The public and their queries. *JASIST*, 52(3):226–234, 2001.

[8] W. J. Wilbur and L. Coffee. The effectiveness of document neighboring in search enhancement. *IPM*, 30(2):253–266, 1994.

Integrating the "Deep Web" With the "Shallow Web"

Michael Stonebraker
MIT

Public web integration services such as Google and Yahoo provide access to the "shallow web", i.e. those sites that are reachable by a text-oriented crawler. Although this service is very useful, it misses large portion of the information available on the web. Specifically, it misses the "deep web", which is data available behind form-oriented user interfaces. Examples of deep web sites include all airline sites, 411.com, weather.com, and most retail sites. These all require one to issue a query through a form-oriented interface to an underlying data base. Obviously, current crawlers are incapable of accessing the deep web. In this paper, we propose a methodology for integrating the deep web with the shallow web.

Our proposal builds on a research prototype, Morpheus, which we have built over the last two years at MIT and the University of Florida. Morpheus is focused on enterprise data integration and database schema heterogeneity. For example, a multinational corporation would have employees in several countries, each with a local salary. Hence, the French employee data base would record French salaries in Euros, after taxes, and including a lunch allowance. In contrast, the US employee data base would contain salary records, gross in dollars with no lunch allowance. Obviously, retrieving the two collections of salaries will produce garbage. Hence, a global schema must be defined and the local schemas must be mapped to this global schema.

There are many approaches to this issue of schema integration, including forcing standardization, using description languages such as Owl, and performing automatic translation based on some sort of logic description of the source and the target objects. Our point of view is that a transform, written in some programming notation, is generally required to convert between enterprise data elements, which we call data types. The objective of Morpheus is to facilitate building and reusing such transforms.

As such, Morpheus is a software system that contains a metadata repository about transforms and data types, a sophisticated browser that allows a human to find objects of interest in the repository and a high level transform construction tool (TCT) that allows a human to build transforms from scratch as well as "morph" existing ones into a form that meets his needs. Morpheus is built on top of Postgres; hence transforms are Postgres user-defined functions. As such, transforms can be called by running Postgres queries on input data stored in Postgres, producing output data in other Postgres tables. As a result of this architecture, web services can be "wrapped" to convert them to user-defined functions, allowing Morpheus to interact with remote data and services.

We are working on several Morpheus extensions, which will allow Morpheus to perform the desired integration of the shallow and deep web:

1) We are building a crawler that will explore the public web, looking for Javascript forms. We plan to semi-automatically wrap each such site to turn it into a Morpheus transform and register it in the Morpheus repository. Human involvement is required to ascertain how (if at all) the new transform is related to existing Morpheus objects.

   This will extend Morpheus to know about the subset of the deep web that is behind Javascript forms

2) It is straightforward to model our metadata tables as RDF objects. Hence, we can build a mapping layer that will support Sparkl access to our Postgres repository.

   This will support Sparkl access to the deep web through Morpheus.

3) We will build a GUI that will support a simple query interface and convert it to Sparkl. Using this interface, an end user could, for example, type professor("Mike Stonebraker")

4) We will build a run time system that performs the following actions. It will decide which of the Javascript wrappers to invoke, run them, converts the result into RDF and load the result into a temporary Postgres table, modeling RDF objects in a table in a straightforward manner. Any returned object can be "enriched" by passing it through any local Morpheus transforms that would convert the object to another Morpheus data type. Furthermore, it will submit the same query to Google, pass the top X results through a natural language parser to find "subject-verb-object" triples, and load the results into the same Postgres table.

5) The result is a collection of RDF objects from the shallow and deep web that deal with the query: professor("Mike Stonebraker")

6) A graph of the result of 5) will be presented to the user, perhaps through Haystack or some other means. The user can browse this representation. Alternately he can refine the result using Sparkl directly.

Jointly with researchers at the University of Florida, we are building out this prototype. Research support is available to one or more interested MIT students.

Multimodal Question Answering for Mobile Devices

Tom Yeh
Vision Interface Group

INTRODUCTION

In recent years, community-based question answering (QA) services, such as Yahoo Answers! and Naver, have enjoyed growing popularity. These services operate web sites to invite anyone to post open-ended questions for free. The questions can be on a variety of topics ranging from car buying tips to dating advice. They are viewed by millions of users in an online community, some of whom may happen to possess the right expertise to give satisfactory answers to them. Users frequent these sites to seek answers about topics they know little of; at the same time, they often volunteer answers about other topics they happen to be familiar with. It is in the spirit of such reciprocity that people are willing to contribute their expertise and knowledge for the benefits of the whole, without seeking any form of monetary compensation.

Over time, these community-based question answering services have accumulated a considerable amount of human knowledge, which can be readily tapped into with an intelligent search engine. In Yahoo Answers!, when a user asks a question, a search engine can quickly lists previously asked questions that are relevant to the new question. The automatic search mechanism allows the user to receive immediate feedback instead of having to wait for some human expert to answer it. If nothing the search engine provides is satisfactory, the question can still be left as as an open question for the whole community to solve.

However, these QA services have some fundamental limitations that can be improved upon. The first limitation is that conventional QA services provide only a single input modality—text. Sometimes we may find it difficult to phrase a question about an unfamiliar object identifiable only by its distinct visual features. For example, we may see an exotic bird outside of a window and want to know about it. Not knowing how to name the bird, our only resort is to actually describe the bird in our question based on its appearance, such as *a red-feathered, large-beak, round-eye bird.* The second limitation is that most QA services were designed primarily for the desktop platform. Yet, we often encounter new and interesting things around us that arouse our curiosity and instill questions in our mind. Thus, there is a need to support situated and pervasive question answering from a mobile platform, so that we can ask questions as soon as we become curious about something.

Therefore, in this paper, we propose a multi-modal question answering system designed for mobile users. As a solution to the limitation of text-only QA, we introduce an additional modality—photo—to be used directly as part of a multi-modal query that includes both visual and textual cues. Using our system, whenever something in the environment catches a user's attention, the user can learn about it immediately, simply by taking a photo of and asking any question about it.

Unlike conventional question answering services, ours is multi-modal (photo and question) rather than text-only, mobile rather than desktop-centric. Unlike location-based information services such as those based on GPS or RFID, ours is active (users decide when and what to inquire) rather than passive (users receive whatever the system decides to show based on the location cue). Moreover, the idea of using photos directly as queries to an information retrieval system has been explored by various works such as a system that recognizes flowers [Flower] and another system that recognizes fish [Fish] based on photos taken by camera phones. Unlike these photo-based information services, ours is open-ended (users can ask about any topic) instead of domain specific.

SYSTEM ARCHITECTURE

This section gives a high-level overview of the architecture of our multi-modal information retrieval system. The architecture we propose consists of five major components described as follows:

Mobile User Interface To let users issue multi-modal queries from mobile devices, we need to design an interface for users to take photos and enter questions that can be embedded in the queries. This interface also allows users to submit queries to a server and to view relevant questions and answers returned by the server.

Expert Community To answer multi-modal queries users submitted, we rely on a community of experts willing to share their knowledge and expertise with others. The success of existing community-based question answering services ,such as Yahoo Answers! and Naver, has demonstrated that an incentive model based on reciprocity and reputation is sufficient for solicit-

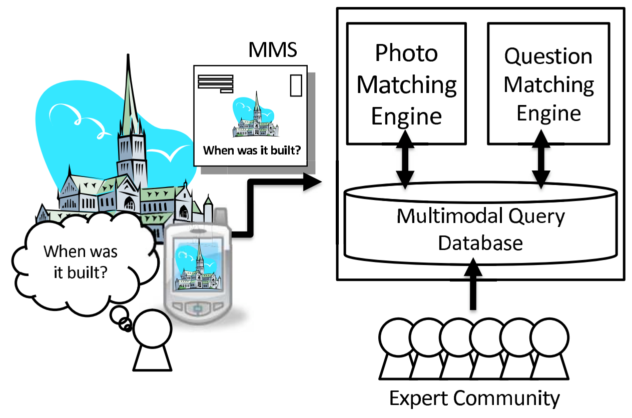


Figure 1. A typical usage scenario (left) and our proposed system architecture (right) of a multi-modal question answering system.

ing altruistic behaviors from community users. It is quite likely the same model that has made these services so successful can be extended to our case.

Multi-modal Query Database To store multi-modal queries ever submitted so that they can be looked up in the future, we need to build a suitable database. Also must be stored in this database are the answers provided by the community users. In effect, this database represents the collective multi-modal knowledge contributed by the community over time. When a multi-modal query is submitted, it will be matched against the queries stored in this database to look up relevant information, before the expert community is consulted.

Photo-matching Engine To check whether someone else has taken a relevant photo in the past, we need a photo-matching engine that can match a new photo against the photos in the database to identify those that are visually similar. Because people can take photos of the same object or scene in various ways, the photo-matching engine needs to provides robustness to image variations due to scale, translation, rotation, and occlusion.

Question-matching Engine To check whether someone else has asked a relevant question in the past, we need a question-matching engine that can match a new question against the questions in the database to identify those that are semantically related.

ALGORITHM

This section describes an algorithm that takes a multi-modal query and lookups relevant queries in a database. The basic idea of the algorithm is to use the photo-matching engine to retrieve a list of candidate photos similar to the photo in the query, and use the question-matching engine to re-rank the candidate photos based on the similarities between the question in the query and the question corresponding to each candidate photo.

Formally, let $x : \langle p, q \rangle$ denote a multi-modal query that consists of a photo p and a question q. Let $X_n = \{x_1 \dots x_n\}$ denote a multimodal database that has seen n queries. The operation of the photo-matching engine can be denoted as:

$$\boldsymbol{f}_i : \langle p, P \rangle \rightarrow \bar{p},$$

whose job is to find finds the i-th most similar photo $\bar{p} \in P$ of a photo p. Similarly, the operation of the question-matching engine can be denoted as

$$\boldsymbol{g}_i : \langle q, Q \rangle \rightarrow \bar{q},$$

whose job is to find the i-th most similar question $\bar{q} \in Q$ of a question q.

Given a new multi-modal query $x_{n+1} : \langle p^*, q^* \rangle$, the algorithm to find the k most relevant multi-modal queries proceeds in three steps:

First, we find a set of candidate photos C_p that consists of the k photos most similar to the query photo p^*:

$$C_p = \{p_i | p_i \leftarrow \boldsymbol{f}_i(p^*, P_n) \ , \ i = 1 \dots k\} \qquad (1)$$

where P_n is the set of n photos already in the database X_n.

Next, we find a set of candidate questions C_q that consists of the questions submitted together with the candidate photos in the same multi-modal queries:

$$C_q = \{q_i | \langle q_i, p_i \rangle \in X_n \ , \ i = 1 \dots k\} \qquad (2)$$

Finally, as results, we return a set of relevant multi-modal queries R, ranked by similarity to the new question:

$$R = \{\langle p_j, q_i \rangle | \langle p_j, \boldsymbol{g}_i(q*, C_q) \rangle \in X_n \ , \ i = 1 \dots k\} \qquad (3)$$