Information Seeking with Social Signals: Anatomy of a Social Tag-based Exploratory Search Browser

Ed H. Chi, Rowan Nairn
Palo Alto Research Center
3333 Coyote Hill Road, Palo Alto, CA 94304 USA
{echi,rnairn}@parc.com

ABSTRACT

Whereas for the fact-retrieval searches, optimal paths to the documents containing the required information are crucial, learning and investigation activities lead to a more continuous and exploratory process with the knowledge acquired during this "journey" being essential as well. Therefore, information seeking systems should focus on providing cues that might make these explorations more efficient. One possible solution is in building information seeking systems in which navigation signposts are provided by social cues provided by a large number of other people. One possible source for social cues is all of the social bookmarks on social tagging sites. Social tagging arose out of the need to organize found information that is worth revisiting. The collective behavior of users who tagged contents seems to offer a good basis for exploratory search interfaces, even for users who are not using social bookmarking sites. In this paper, we present the algorithm of a tag-based exploratory system based on this idea.

Author Keywords
Social Tagging, Exploratory Interfaces, Social Search

ACM Classification Keywords
H3.3 [Information Search and Retrieval]: Relevance Feedback, Search Process, Selection Process; H5.2. [Information interfaces and presentation]: User Interfaces

INTRODUCTION

Existing search engines on the Web are often best on search tasks that involve finding a specific answer to a specific question. However, users of web search engines often need to explore new topic areas, specifically looking for general coverage of a topic area to provide an overview. As part of information seeking, these kinds of exploratory searches involves ill-structured problems and more open-ended goals, with persistent, opportunistic, iterative, multi-faceted processes aimed more at learning than answering a specific query [18, 23].

One existing solution to exploratory search problems is the use of intelligent clustering algorithms that groups and organizes search results into broad categories for easy browsing. Clusty from Vivisimo (clusty.com) is one relatively successful example of these kinds of systems that grew out of AI research at Carnegie Mellon University. One well-known example is Scatter/Gather, which used fast clustering algorithm to provide a browsing interface to very large document collections [6]. These efforts can be seen as continuing a long line of search system research on user relevance feedback, which is a set of techniques for users to have an interactive dialog with the search system, often to explore a topic space [2, 21]. These clustering-based browsing systems extract patterns in the content to provide grouping structures and cues for users to follow in their exploratory searches, often narrowing down on more specific topic areas or jumping between related sub-topics.

Several researchers have suggested the possibility of aggregating social cues from social bookmarks to provide cues in social search systems [13]. We wish to seriously explore the use of social cues to provide navigational aids to exploratory users. The problem with freeform social tagging systems is that, as the tagging systems evolve over time, their information signal declines and noise increases, due to synonyms, misspellings, and other linguistic morphologies [4].

We designed and implemented a tag-based exploratory search system called MrTaggy.com, which is constructed with social tagging data, crawled from social bookmarking sites on the web. Based on the TagSearch algorithm, MrTaggy performs tag normalizations that reduces the noise and finds the patterns of co-occurrence between tags to offer recommendations of related tags and contents [15]. We surmised that the related tags help deal with the vocabulary problem during search [10]. The hope is that these recommendations of tags and destination pages offer support to the user while exploring an unfamiliar topic area.

In a recent paper [15], we described a user experiment in which we studied the learning effects of subjects using our tag search browser as compared against a baseline system.

Workshop SRS'10, February 7, 2010 Hong Kong, China

We found that MrTaggy's full exploratory features provide to the users a kind of scaffolding support for learning topic domains, particularly compensating for the lack of prior knowledge in the topic area.

However, due to space limitations, the previous paper did not present any details on the implementation and algorithm design of the system. In this paper, we detail the design and implementation of MrTaggy.com.

First, we briefly give an overview of the overall user interface and system. Then we focus specifically on a deeper discussion of the design choices we made in the system, as well as the MapReduce architecture needed to model and process 140 million bookmarks using a probabilistic graph model. We then provide a quick overview of the user study reported previously, and finally offer some concluding remarks.

RELATED WORK

Much speculation in the Web and the search engine research community has focused on the promise of "Social Search". Researchers and practitioners now use the term "social search" to describe search systems in which social interactions or information from social sources are engaged in some way [7]. Current social search systems can be categorized into two general classes:

(1) **Social answering systems** utilize people with expertise or opinions to answer particular questions in a domain. Answerers could come from various levels of social proximity, including close friends and coworkers as well as the greater public. Yahoo! Answers (answers.yahoo.com) is one example of such systems. Early academic research includes Ackerman's Answer Garden [1], and recent startups include Aardvark (vark.com) and Quora (quora.com).

Some systems utilize social networks to find friends or friends of friends to provide answers. Web users also use discussion forums, IM chat systems, or their favorite social networking systems like Facebook and Twitter to ask their social network for answers that are hard to find using traditional keyword-based systems. These systems differ in terms of their immediacy, size of the network, as well as support for expert finding.

Importantly, the effectiveness of these systems depends on the efficiency in which they utilize search and recommendation algorithms to return the most relevant past answers, allowing for better constructions of the knowledge base.

(2) **Social feedback systems** utilize social attention data to rank search results or information items. Feedback from users could be obtained either implicitly or explicitly. For example, social attention data could come from usage logs implicitly, or systems could explicitly ask users for votes, tags, and bookmarks.

Many researchers in the information retrieval community have already explored the use of query logs for aiding later searchers [20, 8, 11]. Direct Hit[1] was one early example from early 2001 that used click data on search results to inform search ranking. The click data was gathered implicitly through the search engine usage log.

Others like Google's SearchWiki are allowing users to explicitly vote for search results to directly influence the search rankings. Indeed, vote-based systems are becoming more popular recently. Interestingly, Google's original ranking algorithm PageRank could also be classified as an implicit voting system by essentially treating a hyperlink as a vote for the linked content.

Popularity data derived from other social cues could also be used in ranking search results. Several researchers in CSCW have noted how bookmarks and tags serve as signals to others in the community. For example, Lee found that analyses of del.icio.us users who perceive greater degrees of social presence are more likely to annotate their bookmarks to facilitate sharing and discovery [17]. A well-known study by Golder and Huberman showed that there is remarkable regularity in the structure of the social tagging systems that is suggestive of a productive peer-to-peer knowledge system [12].

In both classes of social search systems, there are still many opportunities to apply sophisticated statistical and structure-based analytics to improve search experience for social searchers. For example, expertise-finding algorithms could be applied to help find answerers who can provide higher-quality answers to particular questions in social answering systems. Common patterns between question-and-answer pairs could be exploited to construct semantic relationships, which may be used to draw automatic inferences to new questions. Data mining algorithms could construct ontologies that are useful for browsing through the tags and bookmarked documents.

SOCIAL BOOKMARKS AS NAVIGATION SIGNPOSTS

Recently there has been an efflorescence of systems aimed at supporting social information foraging and sensemaking. These include social tagging and bookmarking systems for photos (e.g., flickr.com), videos (e.g., youtube.com), or Web pages (e.g., del.icio.us). A unique aspect of tagging systems is the freedom that users have in choosing the vocabulary used to tag objects: any free-form keyword is allowed as a tag.

Tagging systems provide ways for users to generate labeled links to content that, at a later time, can be browsed and searched. Social bookmarking systems such as del.icio.us already allow users to search the entire database for websites that match particular popular tags. Tags can be

[1] http://www.searchengineshowdown.com/features/directhit/review.html

organized to provide meaningful navigation structures, and, consequently, can be viewed as an external representation of what the users learned from a page and of how they chose to organize that knowledge.

Using social tagging data as "navigational advice" and suggestions for additional vocabulary terms, we are interested in designing exploratory search systems that could help novice users gain knowledge in a topic area more quickly.

However, one problem is that the social cues given by people are inherently noisy. Social tagging generates vast amounts of noise in various forms, including synonyms, misspellings, and other linguistic morphologies, as well as deliberate spam [4]. In past research, we showed that extracting patterns within such data becomes more and more difficult as the data size grows [4]. This research shows that an information theoretic analysis of tag usage in del.icio.us bookmarks suggest of decreased efficiency in using tags as navigational aids [4].

To combat noisy patterns in tags, we have designed a system using probabilistic networks to model relationships between tags, which are treated as topic keywords. The system enables users to quickly give relevance feedbacks to the system to narrow down to related concepts and relevant URLs. The idea here is to bootstrap the user quickly with other related concepts that might be gleamed from social usage of related tags. Also, the popularities of various URLs are suggestive of the best information sources to consult, which the user can use as navigational signposts.

THE TAGSEARCH ALGORITHM

Here we describe an algorithm called TagSearch that uses the relationships between tags and documents to suggest other tags and documents. Conceptually, given a particular tag, for tag suggestion, we want to construct a semantic similarity graph as in Figure 1.

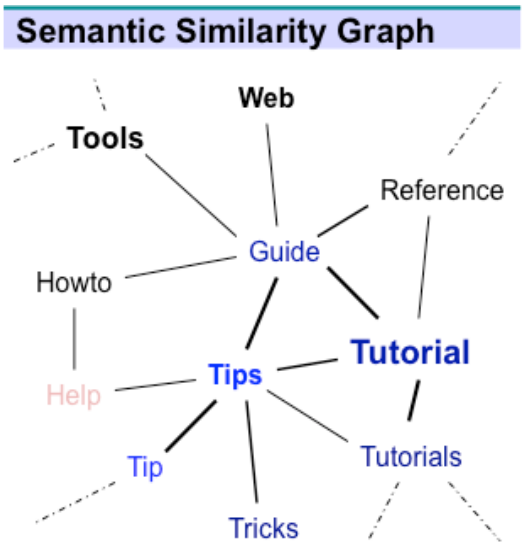

Figure 1: Conceptual Semantic Similarity Relationships between Tags.

Of course, for a URL, if we want to suggest other URLs, we also want to form a similar semantic similarity graph, which is mediated by the tags. There are also the cases where, given a tag, we want to suggest URLs, and vice versa.

In our approach, the idea is to first form a bigraph between document and tagging pairs. Each tagging data in a tuple specifies a linking relationship between a tag and a document object. For each URL, we want to know the probability of a particular tag being relevant to that URL, and vice versa. For a URL, the probability *p(Tag|URL)* can be roughly estimated by the number of times a particular tag is applied by users divided by total number of times all tags are used for a URL. Figure 2 depicts this bi-graph.

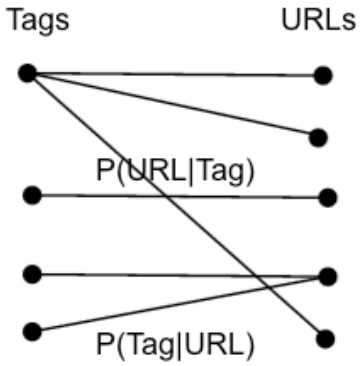

Figure 2: bigraph between document/tag.

For a tag, we want to also compute the probability that a URL is related to it. This probability, p(URL|tag), can be estimated by dividing the number of times an URL is tagged with a particular tag divided by the total number of times the URL is tagged.

A sketch of the idea behind the algorithm is as follows.

To suggest tags:

(1) What we want to do is then form a "tag profile" for a tag, which is the set of other tags that are related to the tag. To compute the tag profiles, we use the bigraph to perform a spreading activation to find a pattern of other tags that are related to a set of tags. Once we have the tag profiles, we can find other tags that are related by comparing these tag profiles. That is, for a given tag, we can compare its tag profile to other tag profiles in the system to find the top most related tags.

(2) Another way to do the same thing is to form a "document profile" for a tag, which is the set of other documents that are related to the tag, similarly using spreading activation. We can then find other tags that are related using these document profiles.

To suggest documents:

(3) We can form "tag profiles" for a document, which is the set of other tags that are related to that document, again using the spreading activation method. We can then

compare these tag profiles for documents to other document tag profiles to find similar documents.

(4) We can form "document profiles" for a document using the spreading activation method over the bigraph. We compare these document profiles for documents to find similar documents.

Steps

Having described the conceptual ideas behind the algorithm, we now turn to the specific steps of the algorithm. TagSearch is done using a multi-step process:

(Step 1) First, we construct a bigraph between URLs and tagging keywords. Bookmarks in these systems are typically of the form [url, tag1, tag2, tag3, tag4, ….]. We can decompose/transform them into the form [url, tag1], [url, tag2], and so on.

Given tuples in the form [url, tag], we can form a bigraph of URLs linked to tags. This bigraph can be expressed as a matrix. This process is depicted in Figure 3.

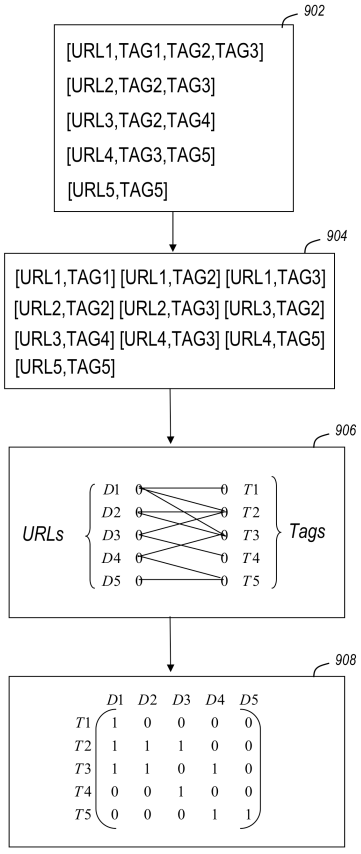

Figure 3: encoding of the tag/document relationships into a bigraph matrix.

(Step 2) Next, we construct "tag profiles" and "document profiles" for each URL and each tag in the system. For each URL and tag in the bigraph, we perform a spreading activation using that node in the bigraph as the entry node. We can do this, for example, using the bigraph matrix constructed in step 1.

After "n" steps (which can be varied based on experimentation), depending on whether the spreading activation was stopped on the tag side of the bigraph or the document side of the bigraph, we will have a pattern of weights on tags or documents. These patterns of weights form the "tag profiles" or "document profiles". This process is depicted in Figure 4.

$$\begin{pmatrix}1\\0\\0\\0\\0\end{pmatrix} = \begin{pmatrix}1\\0\\0\\0\\0\end{pmatrix} \quad n=1$$

$$\begin{array}{c} \\ T1\\T2\\T3\\T4\\T5 \end{array}\!\!\begin{array}{c} \begin{array}{ccccc} D1 & D2 & D3 & D4 & D5 \end{array} \\ \begin{pmatrix}1&0&0&0&0\\1&1&1&0&0\\1&1&0&1&0\\0&0&1&0&0\\0&0&0&1&1\end{pmatrix} \end{array}\!\!\begin{array}{c} A[1] \\ \begin{pmatrix}1\\0\\0\\0\\0\end{pmatrix} \end{array} = \begin{array}{c} A[2]_{TAG} \\ \begin{pmatrix}1\\1\\1\\0\\0\end{pmatrix} \end{array} \quad n=2$$

$$\begin{array}{c} \\ D1\\D2\\D3\\D4\\D5 \end{array}\!\!\begin{array}{c} \begin{array}{ccccc} T1 & T2 & T3 & T4 & T5 \end{array} \\ \begin{pmatrix}1&1&1&0&0\\0&1&1&0&0\\0&1&0&1&0\\0&0&1&0&1\\0&0&0&0&1\end{pmatrix} \end{array}\!\!\begin{array}{c} A[2] \\ \begin{pmatrix}1\\1\\1\\0\\0\end{pmatrix} \end{array} = \begin{array}{c} A[3]_{DOCS} \\ \begin{pmatrix}3\\2\\1\\1\\0\end{pmatrix} \end{array} \quad n=3$$

$$\begin{array}{c} \\ T1\\T2\\T3\\T4\\T5 \end{array}\!\!\begin{array}{c} \begin{array}{ccccc} D1 & D2 & D3 & D4 & D5 \end{array} \\ \begin{pmatrix}1&0&0&0&0\\1&1&1&0&0\\1&1&0&1&0\\0&0&1&0&0\\0&0&0&1&1\end{pmatrix} \end{array}\!\!\begin{array}{c} A[3] \\ \begin{pmatrix}3\\2\\1\\1\\0\end{pmatrix} \end{array} = \begin{array}{c} A[4]_{TAGS} \\ \begin{pmatrix}3\\6\\6\\1\\1\end{pmatrix} \end{array} \quad n=4$$

Figure 4. Spreading Activation of the tag/document bigraph.

Spreading activation have been used in many other systems for modeling concepts that might be related, or to model traffic flow through a website [5]. In this case, we use spreading activation to model tag and concept co-occurrences.

Specifically, the tag profiles and document are computed using spreading activation iteratively as vectors A as follows:

$$A[1] = E;$$

$$A[2] = \alpha M * A[1] + \beta E; \ ....$$

$$A[n] = \alpha M * A[n-1] + \beta E;$$

where:

A[1], A[2], ... A[n] are iteratively computed profile vectors of URLs and tags;

E is a unit vector representing a tag or document entry node;

M is a matrix representation of the bigraph arranged by column or row according to the selected entry node;

α and β are parameters for adjusting spreading activation.

(Step 3) Having constructed these profiles, we now have several options for retrieval. These profiles form the basis for doing similarity computations and lookups for retrieval, search, and recommendations. For example, for a given document, if we want to find more related document to it, we have three options:

(a) For a document, we can simply lookup the corresponding document profile, and pick the top highest weighted documents in that profile and return that set;

(b) We can also use the corresponding document profile or tag profile and compare it against all other document or tag profiles in the system, and find the most similar profiles and return the matching documents;

(c) If the document is not already in the bigraph, we can first use standard information retrieval techniques (for example, cosine similarity of the document word vectors) to find the most similar document that is in our bigraph, and use method (a) or (b) above to find related documents in our bigraph.

If instead, for a given document, we want to look for related tags to it, we can:

(a) Lookup the corresponding tag profile for that document, and choose the top highest weighted tags in that profile and return that set;

(b) Use the corresponding document/tag profile for that tag and compare it against all other document/tag profiles for other tags in the system, and find the most similar profiles and return the matching tags.

(c) If the document is not already in the bigraph, we can first use a standard information retrieval technique to find the most similar document that is in our bigraph, and use method (a) or (b) above to find related documents in our bigraph.

For a given tag, if we want to find related documents or related tags to it, we can again use similar methods (a) or (b) as described above, if the tag already exists in our bigraph. If the given tagging keyword is not in the bigraph, we can first perform a standard keyword search to find the first initial related documents and tags. We can then further refine the result set by the above methods.

Multi-word queries

For any query with multiple tags as the query, the system looks up all of the corresponding document profiles for those tags and simply adds these profiles together to find the most relevant documents. Given that the tag profiles are expressed as spreading activation vectors, addition simply adds the activation values together, creating the equivalent of an OR query on the tag keywords.

For a more strict search simulating an AND query, instead of adding the activation vectors together, we only include a result in the set when the URL has actually been tagged with all of the query keywords.

Negative Keywords

For relevance feedback, the user can tell the MrTaggy system that a tag represents concepts that she is not interested in. To handle these 'negative keywords', the system, instead of adding the corresponding document profile, it subtracts the corresponding document profile from the activation vector.

IMPLEMENTATION BASED ON MAP-REDUCE

Having described the algorithms, we now turn to the

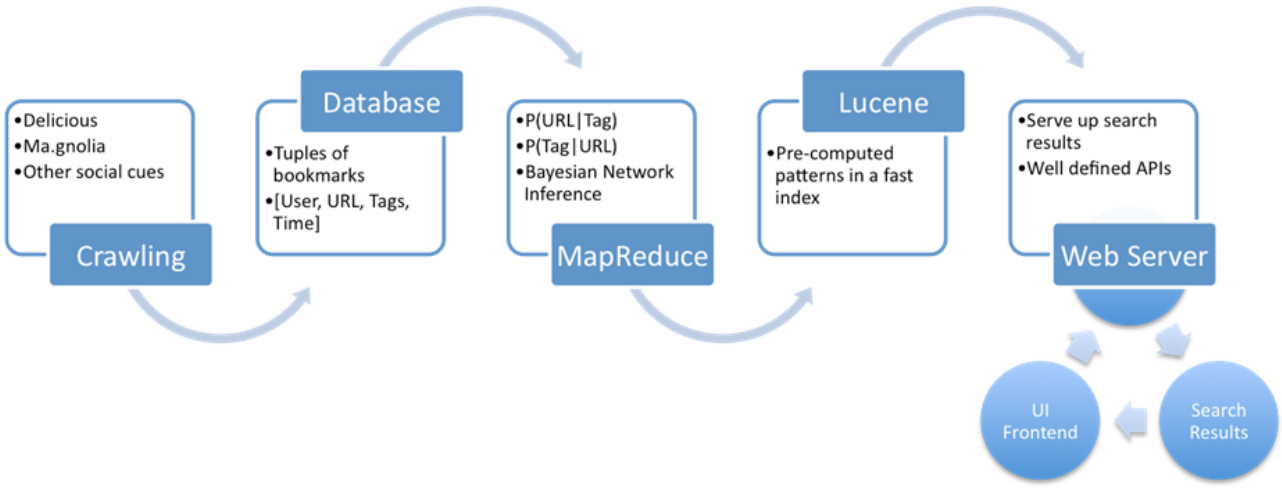

Figure 5. Overall dataflow and architectural diagram of the MrTaggy implementation of the TagSearch algorithm.

description of how we actually implemented the algorithm in a real system. Figure 5 shows an architecture diagram of the overall system we released on the Web called MrTaggy.com.

First, a crawling module goes out to the Web and crawls social tagging sites, looking for tuples of the form <User, URL, Tag, Time>. We keep track of these tuples in a MySQL database. In our current system, we have roughly 140 million tuples.

A MapReduce system based on Bayesian inference and spreading activation then computes the probability of each URL or tag being relevant given a particular combination of other tags and URLs. As described above, we first construct a bigraph between URL and tags based on the tuples and then precompute spreading activation patterns across the graph. To do this backend computation in massively parallel way, we used the MapReduce framework provided by Hadoop (hadoop.apache org). The results are stored in a Lucene index (lucene.apache.org) so that we can make the retrieval of spreading activation patterns as fast as possible.

Finally, a Web server serves up the search results along with an interactive frontend. The frontend responds to user interaction with relevance feedback arrows by communicating with the Web server using AJAX techniques and animating the interface to an updated state.

In terms of data flow, when the user first issues a query, the Web server looks up the related tag recommendations as well as the URL recommendations in the Lucene index and returns the results back to the frontend client.

The client presents the result to the users with the arrows buttons as relevance feedback mechanisms. When the user presses on one of the arrow buttons, the client issues an updated query to the Web server, and a new result set is returned to the client.

MRTAGGY BROWSING/SEARCH INTERFACE

Having just described how the algorithm operates in the backend, we now describe the interaction of the relevance feedback part of the system. Figure 6 shows a typical view of the tag search browser, which is available publicly at MrTaggy.com.

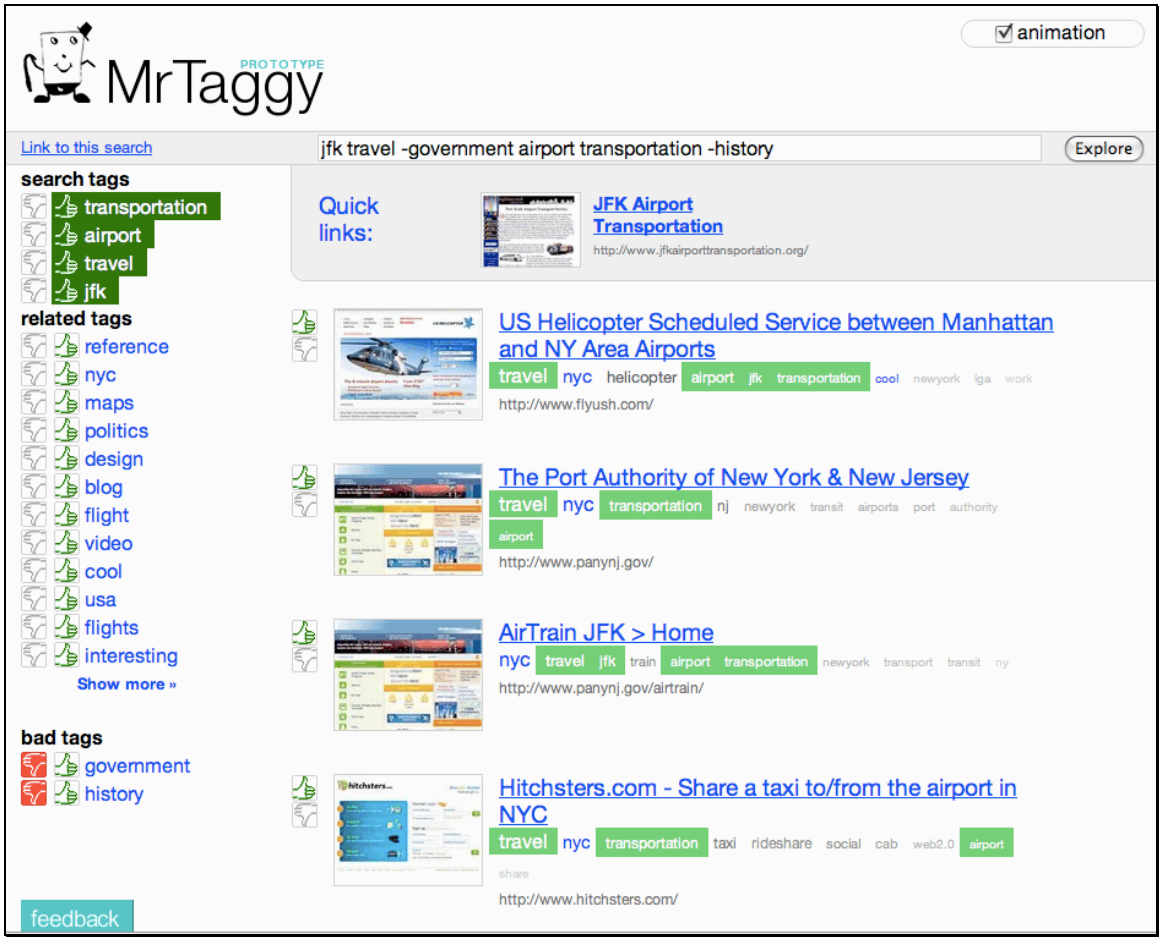

Figure 6. MrTaggy UI with "search tags" section for added tags and "bad tags" section for excluded tags (both on the left).

The left side of the figure shows how the system displays the recommended tags. The right side lists the top document recommendations based on the input so far.

MrTaggy provides explicit search capabilities (search box and search results list) combined with relevance feedback [1, 21] for query refinements. Users have the opportunity to give relevance feedback to the system in two different ways:

Related Page Feedback: By clicking on the downward arrow a search result can be excluded from the results list, whereas by clicking on the upward arrow the search result can be emphasized which leads to an emphasis of other similar Web pages.

Related Tag Feedback: The left of the user interface presentes a *related tags list* (see Figure 6), which is an overview of other tags related to the relevant keywords typed into the search box. For each related tag, up and down arrows are displayed to enable the user to give relevance feedbacks. The arrows here can be used for query refinements either by adding a relevant tag or by excluding an irrelevant one.

In addition, users can refine the search results using tags associated with each of the search results. During search, result snippets (see Figure 7) are displayed in the search results list. In addition to the title and the URL of the corresponding Web page, instead of a short summary description, a series of tags are displayed. Other users have used these tags to label the corresponding Web page. When hovering over tags presented in the snippet, up and down arrows are displayed to enable relevance feedbacks on these tags as well.

Users' relevance feedback actions lead to an immediate reordering or filtering of the results list, since the relevance feedback and the search result list are tightly coupled in the interface. We use animations to display the reordering of the search results, which emphasizes the changes that occurred in the result list (see Video at http://www.youtube.com/watch?v=gwYbonHI5ss). New search results due to the refinements are marked with a yellow stripe.

Quick Links: A recently added feature of the interface is the mashup of quick search results from the Yahoo! BOSS search API. In parallel with the TagSearch process, we issue a Yahoo! Search to obtain the top 1-2 results and display those results as quick links. In this way, if our bookmarks do not offer any good suggestions, the user could use these quick links to get started on their topic explorations as well.

SUMMARY OF THE EVALUATION

We recently completed a 30-subject study of MrTaggy and Kammerer et al. describes the study in detail [15]. In this study, we analyzed the interaction and UI design. The main aim was to understand whether and how MrTaggy is beneficial for domain learning.

We compared the full exploratory MrTaggy interface to a baseline version of MrTaggy that only supported traditional query-based search. In a learning experiment, we tested participants' performance in three different topic domains and three different task types. The results are summarized below:

(1) Subjects using the MrTaggy full exploratory interface took advantage of the additional features provided by relevance feedback, without giving up their usual manual query typing behavior. They also spent more time on tasks and appear to be more engaged in exploration than the participants using the baseline system.

(2) For learning outcomes, subjects using the full exploratory system generally wrote summaries of higher quality compared to baseline system users.

(3) To also gauge learning outcomes, we asked subjects to generate keywords and input as many keywords as possible that were relevant to the topic domain in a certain time limit. Subjects using the exploratory system were able to generate more reasonable keywords than the baseline system users for topic domains of medium and high ambiguity, but not for the low-ambiguity domain.

Our findings regarding the use of our exploratory tag search system are promising. The empirical results show that subjects can effectively use data generated by social tagging as "navigational advice". The tag-search browser has been shown to support users in their exploratory search process. Users' learning and investigation activities are fostered by both relevance feedback mechanisms as well as related tag suggestions that give scaffolding support to domain understanding. The experimental results suggest that users' explorations in unfamiliar topic areas are supported by the domain keyword recommendations and the opportunity for relevance feedback.

CONCLUSION

For exploratory tasks, information seeking systems should focus on providing cues that might make these explorations

Figure 7. The 3 parts of a search result snippet in the MrTaggy interface: title, tags, URL.

more efficient. One possible solution is building social information seeking systems, in which social search systems utilizes social cues provided by a large number of other people. Social bookmarks provide one such exploratory cue that systems can utilize for navigational signposts in a topic space.

In this paper, we described the detailed implementation of the TagSearch algorithm. We also summarized a past study on the effectiveness of the exploratory tool. Since social search engines that depend on social cues rely on data quality and increasing coverage of the explorable web space, we expect that the constantly increasing popularity of social bookmarking services will improve social search browsers like MrTaggy. The results of this project point to the promise of social search to fulfill a need in providing navigational signposts to the best contents.

REFERENCES

1. Ackerman, M. S.;McDonald, D. W. 1996. Answer Garden 2: merging organizational memory with collaborative help. In Proceedings of the 1996 ACM Conference on Computer Supported Cooperative Work CSCW '96. ACM, New York, NY, 97-105.
2. Baeza-Yates, R. and Ribeiro-Neto, B. *Modern Information Retrieval*. Addison Wesley, 1999.
3. Bush, V. As We May Think. *The Atlantic Monthly*, *176,* 1, (1945),101-108.
4. Chi, E.H. and Mytkowicz, T. Understanding the Efficiency of Social Tagging Systems using Information Theory. *Proc. Hypertext 2008*, ACM Press (2008), 81-88.
5. Ed H. Chi, Peter Pirolli, Kim Chen, James Pitkow. Using Information Scent to Model User Information Needs and Actions on the Web. In Proc. of ACM CHI 2001 Conference on Human Factors in Computing Systems, pp. 490--497. ACM Press, April 2001. Seattle, WA.
6. Cutting, Doug, David Karger, Jan Pedersen, and John W. Tukey. Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections. In Proceedings of the 15th Annual International ACM/SIGIR Conference, Copenhagen, 1992
7. Evans, B. and Chi, E. H. Towards a Model of Understanding Social Search. In *Proc. of Computer-Supported Cooperative Work (CSCW)*, pp. 485-494. ACM Press, 2008. San Diego, CA.
8. Fitzpatrick, L. and Dent, M.. Automatic feedback using past queries: social searching? *Proc. 20th Annual Intern. ACM SIGIR Conference*. ACM Press (1997), 306-313.
9. Furnas, G.W., Fake, C., von Ahn, L., Schachter, J., Golder, S., Fox, K., Davis, M., Marlow, C. Naaman, M.. Why do tagging systems work? *Extended Abstracts CHI 2006*, ACM Press (2006), 36-39.
10. Furnas, G.W., Landauer, T.K., Gomez, L.M. and Dumais, S.T.. The vocabulary problem in human-system communication. *Communications of the ACM*, *30* (1987), 964-971.
11. Glance, N.S. Community search assistant. *Proc. IUI 2001*, ACM Press (2001), 91-96.
12. Golder, S. and Huberman, B.A. Usage Patterns of Collaborative Tagging Systems. *Journal of Information Science*, *32,* 2 (2006), 198-208.
13. Heymann, Paul and Koutrika, Georgia and Garcia-Molina, Hector (2008) Can Social Bookmarking Improve Web Search? In: First ACM International Conference on Web Search and Data Mining (WSDM'08), February 11-12, 2008, Stanford, CA.
14. Hong, L., Chi, E.H., Budiu, R., Pirolli, P. and Nelson, L. SparTag.us: Low Cost Tagging System for Foraging of Web Content. *Proc. AVI 2008*, ACM Press (2008), 65-72.
15. Kammerer, Y., Nairn, R., Pirolli, P., and Chi, E. H. 2009. Signpost from the masses: learning effects in an exploratory social tag search browser. In Proceedings of the 27th international Conference on Human Factors in Computing Systems CHI '09. ACM, New York, NY, 625-634.
16. Kittur, A., Chi, E.H., and Suh, B. Crowdsourcing user studies with Mechanical Turk. *Proc. CHI 2008,* ACM Pres (2008), 453-456.
17. Lee, K.J. What Goes Around Comes Around: An Analysis of del.icio.us as Social Space. *Proc. CSCW'06*, ACM Press (2006), 191-194.
18. Marchionini, G. Exploratory search: From finding to understanding. *Communications of the ACM*, *49,* 4 (2006), 41-46.
19. Millen, D., Yang, M., Whittaker, S.and Feinberg, J. Social bookmarking and exploratory search. In L. Bannon, I. Wagner, C. Gutwin, R. Harper, and K. Schmidt (eds.). *Proc. ECSCW'07,* Springer (2007) 21-40.
20. Raghavan, V.V. and Sever, H. On the Reuse of Past Optimal Queries. *Proc. SIGIR95*, ACM Press (1995), 344-350.
21. Shneiderman, B., Byrd, D. and Croft, W.B. Clarifying search: a user-interface framework for text searches, *D-lib magazine*, *3,* 1 (1997), Available at http://www.dlib.org/dlib/january97/retrieval/01shneiderman.html.
22. Simon, H.A. Structure of ill structured problems. *Artificial Intelligence*, *4*, 3-4 (1973), 181-201.
23. White, R.W., Drucker, S.M., Marchionini, M., Hearst, M., schraefel, m.c. Exploratory search and HCI: designing and evaluating interfaces to support exploratory search interaction. *Extended Abstracts CHI 2007*, ACM Press (2007), 2877-2880.