Bookmark Hierarchies and Collaborative Recommendation*

Ben Markines and **Lubomira Stoilova** and **Filippo Menczer**
Department of Computer Science
School of Informatics
Indiana University, Bloomington

Abstract

GiveALink.org is a social bookmarking site where users may donate and view their personal bookmark files online securely. The bookmarks are analyzed to build a new generation of intelligent information retrieval techniques to recommend, search, and personalize the Web. GiveALink does not use tags, content, or links in the submitted Web pages. Instead we present a semantic similarity measure for URLs that takes advantage both of the hierarchical structure in the bookmark files of individual users, and of collaborative filtering across users. In addition, we build a recommendation and search engine from ranking algorithms based on popularity and novelty measures extracted from the similarity-induced network. Search results can be personalized using the bookmarks submitted by a user. We evaluate a subset of the proposed ranking measures by conducting a study with human subjects.

Introduction

Major search engines crawl the Web to populate their databases. When a user submits a query, results are generated and ranked using text similarity measures, the hyperlink structure of the Web, and click-through data from the company's servers. Social bookmarking tools on the other hand build upon the gregarious nature of individuals who establish semantic relationships by sharing URLs. This has led to an explosion of the "folksonomy" phenomenon, as witnessed by the multiplication and popularity of sites such as `del.icio.us` and `citeulike.org`.

Here we describe GiveALink, a system that goes beyond the tagging functionality of current bookmarking sites by actively exploiting collaborative filtering and the hierarchical structure of bookmark files, where present. Hierarchies display a finer, more structured representation of data in comparison to flat tagging systems. The collaborative or social aspect of GiveALink relies on aggregating information from donated bookmark files. Each bookmark file represents a person's notion of semantic similarity. Fig. 1 compares this approach with other recommendation systems.

*We thank T. Holloway, A. Maguitman, L. Rocha, A. Vespignani, R. Henderson, and J. Ratkiewicz for many contributions to the project. Work funded in part by NSF Career Grant IIS-0348940.

Hierarchical Structure	DMOZ	GiveALink
Flat Structure	Classic Content/Link Based	Social Bookmarking
	Single or Shared Representation	Social or Collaborative Aggregation of Users

Figure 1: Two dimensions of recommendation systems: the structure of the knowledge representation space and the social aspect where each individual may contribute a shared or personal representation.

GiveALink distributes the process of collecting data and determining similarity relations among all of its users. We use bookmark files as a convenient existing source of knowledge about what Web pages are important to people, and about the semantic structure in which they are organized. The URLs in our database originate from bookmark files donated and managed by users. We further determine similarity relationships and relevance to queries by mining the structure and attribute information contained in these files. Thus we propose a notion of similarity that is very different from the ones used by Google, Yahoo, and MSN. Our measure of similarity is not based on the content of the pages and not even on the Web link graph. Instead, it is an aggregate of the independent notions of semantic similarity contributed by different bookmark file owners.

Contributions of this work include: (1) A novel semantic similarity measure for URLs that takes advantage of both the hierarchical structure of bookmark files and collaborative filtering techniques. (2) Two ranking measures capturing popularity and novelty, which are based on our similarity measure. (3) An algorithm for personalizing search results based on user bookmarks. (4) Data, such as our (anonymized) URL-to-URL similarity matrix, that is made freely available to the Web community in the hope that it will foster the development of novel and useful Web mining techniques.

Background

Mining Bookmarks Bookmarks are a convenient source of knowledge about the interests of Web users. They are human-edited taxonomies and we have well-established techniques for extracting semantic similarity information from them (Resnik 1995). McKenzie *et al.* (2001) report that people maintain large and possibly overwhelming bookmark collections. Bookmarks are usually highly revisited, but seldom deleted. Abrams *et al.* (1998) suggest that people use bookmarks for different and sometimes unrelated reasons such as fast access, recall, and sharing. We do not make strong assumptions about the way bookmark files are built.

Social Bookmarking Social bookmarking is a way to manage bookmarks for easy access from multiple locations, and also to share them. There are numerous social bookmark sites; several are reviewed by Hammond *et al.* (2005). Users classify bookmarks according to their diverse individual schemas. While traditional search and ranking algorithms allow information producers alone to affect the topic and importance of pages, social bookmarking tools empower information consumers as well. CoWing (Kanawati & Malek 2002) leverages the structure of a bookmark file and collaborative filtering, like GiveALink, but uses a similarity measure based on co-occurance of URLs across folders rather than their hierarchical structure.

Collaborative Filtering In collaborative filtering, patterns in user preferences are mined to make recommendations based on like users' opinions: individuals who have shared tastes in the past will continue to do so. Examples include Ringo (Shardanand & Maes 1995) and GroupLens (Resnick *et al.* 1994) as well as e-commerce sites such as Amazon. Fab (Balabanović & Shoham 1997) combined content-based and collaborative recommendation. Our similarity measure is based on the structure of the bookmarks with no attention to page content. I-Spy (Church, Keane, & Smyth 2004) enhances Web search results based on feedback from community members who submit the same query. Despite their success and popularity, collaborative filtering techniques suffer from some well-known limitations (Sarwar *et al.* 2000): the sparsity of user profiles, the latency associated with precomputing similarity information, and the difficulty in generating predictions about new items. Some of these limitations also apply to the system presented here.

Semantic Similarity Semantic similarity is the degree of relatedness between Web pages, as perceived by humans. Measures of semantic similarity based on taxonomies are well studied (Ganesan, Garcia-Molina, & Widom 2003; Lin 1998). Maguitman *et al.* (2005) extended Lin's (1998) information-theoretic measure to infer similarity from the structure of general ontologies, both hierarchical and non-hierarchical. The ODP (`dmoz.org`) — a human-edited directory of the Web that classifies millions of pages into a topical ontology — can be used as a source of semantic similarity information between pairs of Web sites. Numerous attempts to automate the calculation of semantic similarity between Web pages through observable features, like content and hyperlinks, have been conducted. Surprisingly, measures relying heavily on content similarity (e.g. common words) are poor predictors of semantic similarity (Maguitman *et al.* 2005) while measures that rely mainly on link similarity (common forward/backward edges) estimate semantic similarity with greater accuracy. Here we propose another measure based on combining the tree-based semantic similarity models of a community of users.

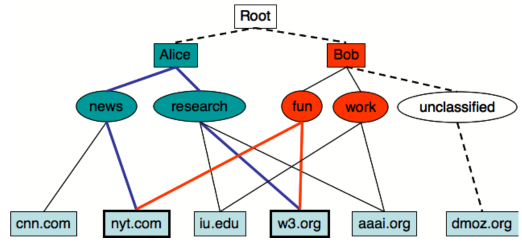

Figure 2: Combining bookmarks with collaborative filtering. Each user has a personal hierarchical representation of links. Here `nyt.com` and `w3.org` have high similarity according to Bob, since they are in the same folder. Alice contributes a smaller (but non-zero) similarity to these URLs, since their lowest common ancestor is the root of her tree.

The GiveALink System

System Architecture

Users can donate their bookmarks anonymously or as registered users at `givealink.org`. We take various precautions to prevent pollution by spammers and to protect the privacy of donors. We require users to pass a CAPTCHA test when donating anonymously and check the MD5 signature of donated files to prevent multiple identical submissions (like default bookmark files). When users register, they have to provide a valid email address. We query the host to make sure that the email address is valid, and then issue the user an activation code. To activate the account, the user has to send an email with their activation code in the subject. We use relay information from the email to verify the source. This registration process protects users from email cluster bomb DDoS attacks (Jakobsson & Menczer 2003).

When users donate bookmarks, we parse their files by determining browser and platform from the user-agent header. Our set of parsers supports Internet Explorer, Netscape, Mozilla, Firefox, Opera, and Safari.[1]

The back end of the system is anchored by a MySQL database server. The data stored in the database includes users, browser and platform data, the directory structure of the bookmark files, the URLs themselves, as well as some personalized information about the URLs such as descriptions that users entered and the time the bookmark was created and last accessed.

Bookmark Similarity

The URLs in a bookmark file are organized in directories and subdirectories and thus have an underlying tree structure. We view the bookmarks submitted by one user as a tree rooted at her username. Fig. 2 illustrates the structure of two user bookmark files.

[1]The latest version of Safari uses binary XML, so we developed a Web service for conversion to ASCII (`homer.informatics.indiana.edu/cgi-bin/plutil/plutil.cgi`).

To exploit the hierarchical structure of bookmark files, we use Lin's (1998) measure to calculate similarity between the URLs in a user u's tree. Let URL x be in folder F_x^u, URL y be in folder F_y^u, and the lowest common ancestor of x and y be folder $F_{a(x,y)}^u$. Also, let the size of any folder F, $|F|$ be the number of URLs in that folder and all of its subfolders. The size of u's root folder is $|R_u|$. Then the similarity between x and y according to user u is:

$$s_u(x,y) = \frac{2\log\left(\frac{|F_{a(x,y)}^u|}{|R_u|}\right)}{\log\frac{|F_x^u|}{|R_u|} + \log\frac{|F_y^u|}{|R_u|}}. \tag{1}$$

This function produces similarity values in $[0, 1]$. If two URLs appear in the same folder, $s_u(x,y) = 1$ because $F_x^u = F_y^u = F_{a(x,y)}^u$. There are two caveats that lead to changes in this basic tree representation. First, consider the two highlighted URLs in Fig. 2 and their similarity according to Alice. Because the lowest common ancestor is Alice's root, $F_{a(x,y)}^{Alice} = R_{Alice}$ and thus $s_{Alice}(x,y) = 0$. However for collaborative filtering we want to capture a minimal association from Alice having bookmarked both URLs. Therefore we add a virtual, global root R and use it in place of user root R_u in Eq. 1, yielding $s_{Alice}(x,y) > 0$. Second, many users keep (some or all) bookmarks unorganized in the top-level folder. This is not to be interpreted as a strong semantic association, so we do not want to assign a maximal similarity of 1 among all unclassified URLs. Therefore we create virtual folders for each unclassified URL. As a result, such URLs will have minimal similarity to each other. Both modifications are illustrated in Fig. 2 with dashed lines.

Lin's measure is only appropriate for calculating the similarity of URL pairs according to a single user. To obtain a global, collaborative similarity measure for URLs x and y, we sum the similarities reported by each user:

$$s(x,y) = \frac{1}{N}\sum_{u=1}^{N} s_u(x,y).$$

If a user has both URLs x and y, then he reports $s_u(x,y)$ according to Equation 1, otherwise he reports $s_u(x,y) = 0$. If a user has URL x in multiple locations, we report the highest value. It is important to point out that here N is the total number of users, not just those with $s_u(x,y) \neq 0$. Thus the more users who share x and y, the higher $s(x,y)$. The final similarity matrix represents a weighted undirected graph where the nodes are URLs and the weight of an edge is the similarity of the two connected URLs.

As of February 14, 2006, GiveALink has collected 996 bookmark files that contribute a total of 60,330 unique URLs. The similarity matrix has a density of 1.3%. Fig. 3 visualizes the topology of the similarity network and its well defined clusters. The topics are clearly representative of the community of early GiveALink adopters, many of whom are affiliated with the our computer science department.

The Recommendation System

The pivotal application of GiveALink is a recommendation system that allows users to explore the bookmark collection.

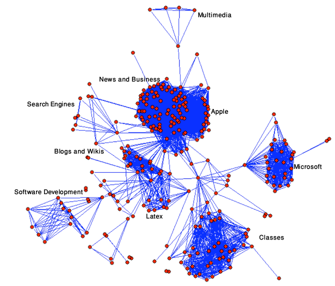

Figure 3: Graph topology generated using Pajek with the top 5,000 similarity edges. Labels are added by hand.

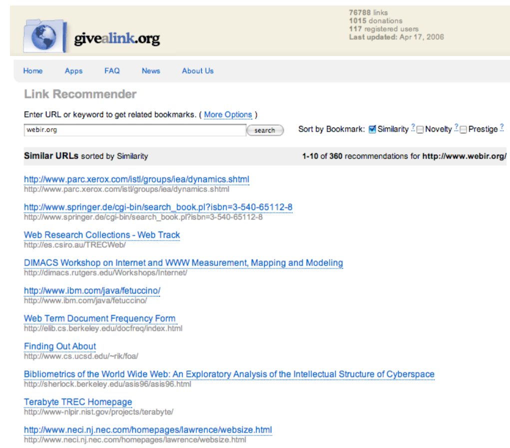

Figure 4: A screen shot of the GiveALink recommendation system, displaying results for the query `webir.org`.

Fig. 4 shows its interface and the results from a simple query. When the user provides a query URL, the system looks for other URLs that have high bookmark similarity to it, according to our matrix s. The user can select several ranking measures (described later), and recommended sites are ranked by the product of their values. By default, results are ranked by similarity to the query.

If the query URL provided by the user is not in the GiveALink database, we resort to help from a search engine to bootstrap the recommendation: we submit the query URL to, say, the Google API and search for similar sites. From the top ten results that Google returns, we pick those that are in our collection and expand the resulting set with additional sites from our database similar to them. We only return URLs that are in our database, and therefore the similarity and ranking values are known for all of them.

We conducted a user study comparing two ranking criteria: GiveALink's similarity s and Google's `related` score. Each subject submitted query URLs and determined whether each resulting URL was relevant or not. From the data collected, precision and recall for each rank were calculated and averaged across all queries.

We included Google in the study for gauging the performance of our collaborative filtering and ranking techniques. Our intention is not to suggest a direct competition with traditional search engines, but rather to set a context in which to interpret the performance of our system. One could in-

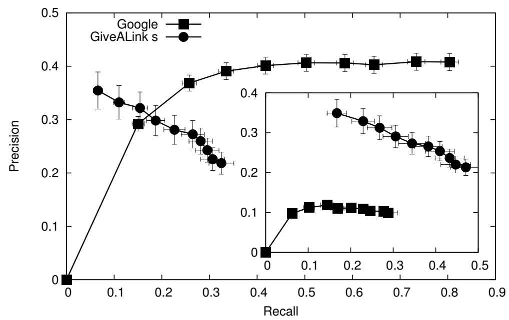

Figure 5: Precision-recall plots for our user study. The data is based on 86 subjects who submitted 336 query URLs and were asked to identify "relevant" pages. Error bars correspond to ± 1 standard error in precision and recall, microaveraged across all user queries for each rank. The reason why Google's curve starts at the origin is that for all user queries Google returned a URL in the first position that was deemed irrelevant. In most cases, this was the query URL itself. The main plot's relevant sets include all URLs from *either* GiveALink's or Google's index that were deemed relevant by the subjects. The inset plot's relevant sets include only relevant URLs in *both* GiveALink and Google.

corporate the GiveALink similarity measure to improve the richer ranking algorithms employed by search engines.

Fig. 5 shows the results of the user study. Recalling that GiveALink does not have access to the pages' content, links, click rate information, or any measure other than the presence of URLs in fewer than a thousand bookmark files, we find it encouraging that the overall performance of GiveALink seems to be comparable with that of Google's `related` service for the top-ranked results. On the other hand the comparison between the performance of GiveALink and Google must be interpreted in light of the enormous difference between the coverage of the two systems — GiveALink's less than $30,000$ URLs (at the time of the experiment) were at least five orders of magnitude less than the number of pages indexed by Google. To factor out the effect of such different coverage on performance, an alternative comparison is offered in the inset of Fig. 5. Here we restrict the relevant sets to only include URLs that, in addition to being deemed relevant, appear in *both* GiveALink's and Google's indices. Under the reasonable assumption that GiveALink and Google contain independent samples of the Web, this leads to a comparison that normalizes for the relative sizes of the systems to focus on the quality of the ranking functions. In this view, GiveALink's similarity measure seems to outperform Google. We are confident that GiveALink's absolute performance can improve considerably as more bookmarks are collected.

Other Ranking Methods

Popularity

In search engines, query-independent importance measures are used in conjunction with query-dependent similarity in ranking results. To explore this idea, we consider here a popularity measure based on the *centrality* of a node in the similarity network. The centrality of a URL is the average of the shortest-path similarities s_{max} between this node and every other node. A URL with high centrality is one that is very similar to all other URLs in our collection. Therefore popular URLs (appearing in many bookmark files) are also more central.

One possible approach is to compute the similarity on a given path as the product of the similarity values along all edges in the path. For example, if URLs x and y are connected by a path $x \rightsquigarrow z \rightsquigarrow y$, where $s(x,z) = 0.5$ and $s(z,y) = 0.4$, then the similarity between x and y on that path is $s(x \rightsquigarrow y) = 0.5 \times 0.4 = 0.2$. Although this approach is rather intuitive, it is leads to a very fast decay in path similarities. In our system, we convert similarity values to distances, then we compute shortest-path distances using Floyd-Warshall's algorithm, and finally we convert these values back into shortest-path similarity values. To convert between similarity and distance values, we use the transformation $dist(x,y) = [1/s(x,y)] - 1$. Thus the closer two URLs, the higher their similarity. The distance along a given path is the sum of the distances along all edges in the path. The shortest-path similarity between two pages is thus defined as

$$s_{max}(x,y) = \left[1 + \min_{x \rightsquigarrow y} \sum_{(u,v) \in x \rightsquigarrow y} \left(\frac{1}{s(u,v)} - 1\right)\right]^{-1}.$$

Computing centrality is time consuming. If the number of URLs in our database is U, then calculating all-pairs shortest path has complexity $O(U^3)$. For a more feasible measure of centrality , we introduce an approximation called *prestige.* Prestige is a recursive measure inspired by PageRank — the prestige of a URL is tied to the prestige of its neighbors in the similarity graph. The difference between prestige and PageRank is that the latter is computed on a directed, unweighted graph where edges represent hyperlinks; prestige is computed on our undirected, weighted similarity graph s. The iterative process is defined as follows: at time $t = 1$, we give all of the URLs prestige values equal to 1. For each consecutive step, the prestige of node i at time $t+1$ is

$$p_i(t+1) = (1-\alpha) + \alpha \cdot \sum_j \frac{s(i,j) \cdot p_j(t)}{\sum_k s(j,k)}.$$

We use $\alpha = 0.85$. The computation continues until the prestige values converge, $p_i = \lim_{t\to\infty} p_i(t)$. We observed a high correlation between prestige and centrality values (Pearson's $\rho = 0.72$), thus prestige serves as a good approximation for centrality. We will refer to prestige as "popularity" in the remainder of our discussion.

Along with GiveALink's similarity s and Google's `related` score, our user study discussed earlier included an evaluation of popularity. The result in Fig. 6 reveals that ranking by a combination of similarity and popularity decreased the performance of the recommendation systems compared to ranking by similarity alone. To interpret this result, Fig. 6 also illustrates how URLs similar to the query may be degraded by their popularity. If a person queries for URLs similar to `u0` ranked by similarity and popularity, `u1`

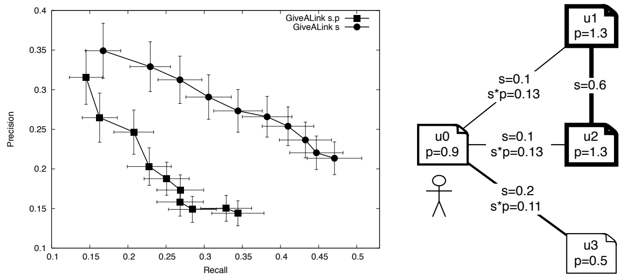

Figure 6: Left: precision-recall plots for ranking by similarity s and by the product $s \cdot p$ of similarity and prestige. The relevant set includes URLs in both GiveALink's and Google's indices. Right: illustration prestige bias (see text).

and `u2` will be ranked higher than `u3` despite `u3` having a higher similarity to `u0`. However in this example `u3` may be a better recommendations. We conclude that while popularity may be useful in some applications, the simple product of similarity and popularity is not a good ranking measure for our recommendation system.

Novelty

For some pairs of URLs the indirect shortest-path similarity s_{max} is higher than the direct edge similarity s. There are pairs of URLs (x, y) where $s(x, y)$ is relatively low, but if both x and y are very similar to a third URL z, then their shortest-path similarity $s_{max}(x, y)$ could be much higher. This violation of transitivity, known as semi-metric behavior (Rocha 2002), is valuable for a recommendation system because it reveals potential similarity that has not yet been discovered by individual bookmark users. If used in addition to a direct similarity measure, it may empower the recommendation system to not only generate natural and obvious suggestions, but also unexpected ones that could inspire users to broaden and deepen their interests. To exploit semi-metric behavior we define a novelty measure:

$$novelty(x, y) = \begin{cases} \frac{s_{max}(x,y)}{s(x,y)} & \text{if } s(x, y) > 0 \\ \frac{s_{max}(x,y)}{s_{min}} & \text{if } s(x, y) = 0 \end{cases}$$

where s_{min} is the smallest non-zero similarity value, $s_{min} = \min_{s(x',y')>0} s(x', y')$. This measure is similar to one of the semi-metric ratios introduced by Rocha (2002). For purposes of recommendation we are only interested in pairs of URLs where $novelty(x, y) > 1$, i.e. the indirect similarity is higher than the direct similarity. Because computing s_{max} is expensive as discussed above, we approximate it by only considering paths of at most two edges, i.e., $s_{max}(x, y) \approx [1 + \min_z(dist(x, z) + dist(z, y))]^{-1}$.

The indirect associations captured by the novelty ratio are a global property of the network and cannot be locally measured from direct association levels (Rocha 2002). If the user chooses to rank search results by novelty (or some combination of measures that include novelty), the recommendations that are non-trivial and unexpected will be ranked higher. Applications of novelty are described next.

Other Applications

Recommendation by Novelty

In the recommendation system, results are generated by mining the database for URLs that have high bookmark similarity to the user query, and they may be ranked by novelty among other measures. A different approach is to only consider novel URLs, i.e. those having $novelty > 1$. The two recommendation systems address different information needs. The former provides additional information that is relevant to the user query; the latter provides only information that relates to the query in a non-trivial way. These results may address the same questions from a different perspective: a different domain of knowledge, perhaps a different time period or geographical location.

Search

Instead of providing a query URL, users also have the option of typing in keywords. The interface of this system mimics the familiar interface of search engines. The query is submitted to a search engine API and the top ten results are matched against the GiveALink database. The URLs retrieved from the database are then ranked by their similarity to the search engine hits. As our bookmark collection grows, our goal is to make the system independent of external search engines. We plan to match the query keywords against the descriptions and titles that users enter in their bookmark files. It would also be possible to crawl and index the donated URLs, although at present this is not a research direction we are pursuing.

Personalization

The GiveALink system allows for search results to be tailored towards the interests of registered users. We calculate a personal similarity score between every URL in our database and the profiles of each registered user, based on how similar the URL is to the user's set of donated bookmarks. There are many possible ways to quantify this similarity measure. One option is to calculate the average similarity between the given URL and the user's bookmarks. Based on preliminary analysis, we believe this approach is not appropriate because a user may have many heterogeneous interests. This would lower the average similarity for relevant URLs.

The measure that we use is the maximum similarity between the given URL and a URL in the user's bookmark collection. If a user profile contains a set of bookmarks B, then the personal similarity between URL x and B is:

$$s_p(x, B) = \begin{cases} \max_{y \in B} s(x, y) & \text{if } x \notin B \\ 1 & \text{if } x \in B. \end{cases} \quad (2)$$

In addition, we would like to pay particular attention to the interests of the user that are unique with respect to the other users. For example, default browser bookmarks should not overly affect the personalization. Thus we weigh the personal similarity by how unlikely it is that the user has a bookmark, in a way analogous to the inverse document frequency in the TFIDF weighting scheme. If the number of GiveALink donations is N and the number of those who own URL y is $N(y)$, we modify Eq. 2 to $s_p(x, B) = \max_{y \in B} [s(x, y) \cdot \log(N/N(y))]$ if $x \notin B$.

The personalized similarity measure is expensive: if U is the number of URLs in our database, B is the number of URLs in the largest bookmark collection donated by a user, and N is the number of users, then the algorithm has complexity $O(U \cdot B \cdot N)$. We precompute the personalized similarity scores for all registered users and store them in the database. The personalized similarity score is then treated as another ranking measure: the recommended results are ranked by the product of query similarity, personal similarity, and any other ranking measures the user selected.

Other Services

To make GiveALink data more accessible, a few additional applications are available. An RSS feed returns GiveALink's results in XML format. Users may either treat this as a Web service or as a channel for related URLs that can be ordered using any of the ranking measures. Also available through the feed are the most popular URLs. A bookmark manager is an interface for users to manage and organize the bookmarks in a personal directory. Finally a bookmarklet allows users to donate individual links.

Conclusions

GiveALink is a public site where users donate their bookmarks to the Web community. The proposed similarity measure for URLs takes advantage of both the hierarchical structure of bookmark files and collaborative filtering across users. The social bookmark network induced by the similarity measure seems to display meaningful clusters. We introduced recommendation systems with popularity and novelty ranking measures extracted from the similarity data. Recommendations can be personalized based on the user bookmarks. We reported on a human subject study confirming that our similarity measure provides an effective way to generate and rank recommendations. One could combine these measures with other criteria from content and link analysis to obtain richer and more effective models of relevance.

An advantage of our system is that we can calculate similarity and make guesses about the topic of a page without having to crawl it. Traditional search engines use text analysis tools (like cosine similarity) to estimate the relevance of a URL with respect to the user query. Our similarity measure does not depend on the content of the page and thus we can recommend URLs for files in various formats, multimedia, and so on without needing access to their content.

Regarding coverage, we note that not all the URLs in our collection are known to Google. We suspect that some users bookmark pages that are not linked from other pages on the Web and thus are invisible to search engine crawlers.

Here we have compared GiveALink with a non-social recommendation system; another important evaluation will involve a direct comparison with tag-based social bookmarking systems. E.g., `del.icio.us` recently introduced hierarchical tags and a "related tag" functionality; a "related URL" feature would be easy to implement. Further efforts will focus in the following directions: (1) The use of popularity in ranking. (2) Widgets and browser plug-ins to broaden the availability of GiveALink's features. (3) Evaluating the novelty and personalization engines. (4) AI techniques for visualizing and navigating the similarity network.

We make all non-personal data freely available to the research community in the hope that it will foster the development of novel and useful Web mining techniques. Our similarity matrix, as well as prestige scores for all bookmarks in our collection, can be downloaded at `givealink.org`.

References

Abrams, D.; Baecker, R.; and Chignell, M. H. 1998. Information Archiving with Bookmarks: Personal Web Space Construction and Organization. In *Proc. ACM CHI'98 Conf.*, 41–48.

Balabanović, M., and Shoham, Y. 1997. Content-based, collaborative recommendation. *Comm. ACM* 40(3):67–72.

Church, K.; Keane, M. T.; and Smyth, B. 2004. The First Click is the Deepest: Assessing Information Scent Predictions for a Personalized Search Engine. In *Proc. AH2004 Workshop*, 173–182.

Ganesan, P.; Garcia-Molina, H.; and Widom, J. 2003. Exploiting Hierarchical Domain Structure to Compute Similarity. In *ACM Trans. Inf. Syst. 21(1)*, 64–93.

Hammond, T.; Hannay, T.; Lund, B.; and Scott, J. 2005. Social Bookmarking Tools (I): A General Review. *D-Lib Magazine* 11(4).

Jakobsson, M., and Menczer, F. 2003. Untraceable email cluster bombs: On agent-based distributed denial of service. *;login:* 25–28.

Kanawati, R., and Malek, M. 2002. A multi-agent system for collaborative bookmarking. In *Proc. AAMAS*, 1137–1138.

Lin, D. 1998. An information-theoretic definition of similarity. In *Proc. 15th ICML*, 296–304. Morgan Kaufmann.

Maguitman, A.; Menczer, F.; Roinestad, H.; and Vespignani, A. 2005. Algorithmic Detection of Semantic Similarity. In *Proc. 14th Intl. WWW Conf.*, 107–116.

McKenzie, B., and Cockburn, A. 2001. An Empirical Analysis of Web Page Revisitation. In *Proc. of the 34th Hawaii International Conference on System Sciences*.

Resnick, P.; Iacovou, N.; Suchak, M.; Bergstorm, P.; and Riedl, J. 1994. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proc. ACM Conf. on Computer Supported Cooperative Work*, 175–186.

Resnik, P. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *IJCAI*, 448–453.

Rocha, L. M. 2002. Semi-metric behavior in document networks and its application to recommendation systems. In Loia, V., ed., *Soft Computing Agents: A New Perspective for Dynamic Information Systems*, 137–163. IOS Press.

Sarwar, B. M.; Karypis, G.; Konstan, J.; and Riedl, J. 2000. Analysis of recommender algorithms for e-commerce. In *Proc. 2nd ACM E-Commerce Conf. (EC'00)*.

Shardanand, U., and Maes, P. 1995. Social Information Filtering: Algorithms for Automating "Word of Mouth". In *Proc. ACM CHI'95 Conf.*, 210–217.