GiveALink: Mining a Semantic Network of Bookmarks for Web Search and Recommendation

Lubomira Stoilova
Comp. Sci. Dept.
Indiana University
lstoilov@cs.indiana.edu

Todd Holloway
Comp. Sci. Dept.
Indiana University
tohollow@cs.indiana.edu

Ben Markines
Comp. Sci. Dept.
Indiana University
bmarkine@cs.indiana.edu

Ana G. Maguitman
School of Informatics
Indiana University
anmaguit@cs.indiana.edu

Filippo Menczer
Informatics & Comp. Sci.
Indiana University
fil@indiana.edu

ABSTRACT

GiveALink is a public site where users donate their bookmarks to the Web community. Bookmarks are analyzed to build a new generation of Web mining techniques and new ways to search, recommend, surf, personalize and visualize the Web. We present a semantic similarity measure for URLs that takes advantage both of the hierarchical structure of the bookmark files of individual users, and of collaborative filtering across users. We analyze the social bookmark network induced by the similarity measure. A search and recommendation system is built from a number of ranking algorithms based on prestige, generality, and novelty measures extracted from the similarity data.

Keywords

Semantic Similarity, Collaborative Filtering, Web Search, Social Bookmark Networks

1. INTRODUCTION

The GiveALink project is an attempt to explore alternatives to centralized search algorithms. Traditional search engines today crawl the Web in order to populate their database. 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.

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. All of the URLs in our database originate from bookmark files donated by users. We further determine similarity relationships and relevance to queries by mining the structure and the 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 in our database 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.

There are several other Web sites that collect bookmarks and provide services such as sharing, tagging, and full-text search. These include `Del.icio.us`, `Simpy`, `Furl`, `Spurl`, `Backflip`, `CiteULike`, and `Connotea` and are reviewed by Hammond et al. [6]. GiveALink is different in that we actively exploit both collaborative filtering and the hierarchical structure of bookmark files, where present. We develop novel Web mining techniques and applications that are not already available elsewhere. Furthermore, GiveALink is a non-commercial research project and both our data and algorithms are openly available to the Web community.

2. BACKGROUND

The GiveALink system collects donated bookmark files and applies collaborative filtering techniques to them to estimate the semantic similarity between the bookmarked URLs. This section introduces definitions and some previous work related to collaborative filtering and semantic similarity. It also discusses briefly the type of information contained in bookmark files.

2.1 Collaborative Filtering

Collaborative filtering, also referred to as social information filtering, identifies and exploits common patterns in the preferences of users. Traditionally, it has been used to identify communities and build recommendation systems based on like users opinions. Examples include Ringo [16] for personalized music recommendations and GroupLens [12] for filtering streams of net news, as well as e-commerce sites such as Amazon.com [17] that make personalized product recommendations. These systems are predicated on the assumption that individuals who have shared tastes in the past will continue to share tastes in the future.

Collaborative filtering techniques are also used for inferring global structures in information domains. A prominent example is Page-Rank [3], a global measure of citation importance for URLs. To a first-degree approximation, PageRank assumes that the number of citations or inlinks to a Web page is a testimony for its importance and quality. Of course, the hyperlink structure of the Web has been created by individual users adding links from their pages to other pages. Thus the count of citations to a Web page is in essence a collaborative filtering measure.

Despite their success and popularity, collaborative filtering techniques suffer from some well-known limitations [15]. One is the sparsity of user profiles: the number of items contained in one user

LinkKDD'05, August 21, 2005, Chicago, IL, USA

profile is negligible compared to the entire dataset (the Web in our case). Thus the information contributed by one user is small and it is hard to infer communities because there is little overlap between profiles. Another critical limitation is the complexity of collaborative filtering algorithms. The latency associated with processing large data sets requires that similarity information is pre-computed offline. When a user submits a new profile, the data is not integrated into the system until the next time the database is rebuilt. The user profile is not updated until then either. Finally, collaborative filtering systems cannot generate predictions about new items. Since similarity is estimated based on existing user profiles, the system cannot generate recommendations for URLs that the users are not already familiar with. Many of these limitations also apply to the system described here.

2.2 Semantic Similarity

Semantic similarity between Web sites is a term used to describe the degree of relatedness between the $meanings$ of the Web sites, as it is perceived by human subjects. Measures of semantic similarity based on taxonomies (trees) are well studied [5, 8]. Recently Maguitman *et al.* [9] have extended Lin's [8] information-theoretic measure to infer similarity from the structure of general ontologies, both hierarchical and non-hierarchical. The ODP[1] — 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.

Search engine designers and researchers have made numerous attempts to automate the calculation of semantic similarity between Web pages through measures based on observable features, like content and hyperlinks. Studies conducted by Menczer and colleagues [9, 11] report, quite surprisingly, that measures relying heavily on content similarity (e.g. common words) are very poor predictors of semantic similarity. On the other hand, measures that only take into consideration link similarity (common forward and backward edges), or scale content similarity by link similarity, estimate semantic similarity with greater accuracy. Incidentally, neither content nor link similarity alone is a good approximation of semantic similarity, and they are also not strongly correlated with each other.

Thus the question of how to automate the semantic similarity measure for arbitrary pages remains open. Here we propose another measure that is based on combining the semantic similarity notions of a community of users through collaborative filtering techniques, and compare it with previously studied measures based on content, links, and the ODP.

2.3 Mining Bookmarks

An issue that we need to consider before looking at the GiveALink system is the type of information we can expect to find in bookmark files. Bookmarks are a convenient source of knowledge about the interests of Internet users. On one hand, they are human-edited taxonomies and we have well-established techniques for extracting semantic similarity information from them. Additionally, they are designed to be easily transferable between browsers and computers and that makes it easy for users to access them and upload them to our Web site. We have considered using other sources of information, like browsing history files, which arguably contain more data. They present some technical and privacy challenges and may be considered at a later time.

Bookmark files contain a mix of explicit and implicit knowledge. The following attributes are explicit in the bookmark file: (1) URLs, (2) titles, (3) the hierarchical structure of the file, and (4) the browser and platform. Additionally, some browsers provide the time when bookmarks are added and last accessed, as well as personalized title and description that users can edit themselves.

We also exploit some of the implicit knowledge contained in bookmarks by taking into consideration the way people generally use these files. McKenzie *et al.* [10] report that people maintain large, and possibly overwhelming, bookmark collections. The bookmarked URLs are usually highly revisited, but they are seldom deleted and often some of the bookmarks are stale links. Additionally, Abrams *et al.* [1] suggest that people use bookmarks for different and sometimes unrelated reasons: some users bookmark URLs for fast access; other users bookmark URLs with long names that they cannot remember; yet others use bookmarks as a way to share their informational space with the community. Thus it is important to not make strong assumptions about the way the bookmark files were built when mining information from them.

3. THE GIVEALINK SYSTEM

This section presents the architecture of the GiveALink donation system and database. It also describes how we mine the collected bookmark files to build a URL-to-URL matrix containing semantic similarity values.

3.1 System Architecture

Users can donate their bookmarks anonymously or as registered users at `givealink.org`. To protect the database from bots that pollute it with a large quantity of engineered bookmark files, we require users to pass a CAPTCHA test [18] when donating anonymously. In addition, we prevent multiple submissions of identical files (like default bookmark files) by checking the MD5 signature of every donated file.

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 to a special email address and include their activation code in the subject of the email. We use relay information from the email to verify that the email is coming from the correct source. This registration process is proposed by Jakobsson and Menczer [7] as an alternative to the double-opt in protocol to avoid email cluster bomb DDoS attacks.

When users donate bookmarks at `givealink.org`, we use their user agents to determine which browser and platform they are using in order to parse the file correctly. Our set of parsers supports Internet Explorer, Netscape, Mozilla, Firefox, Safari, and Opera. The file formats are as follows: Netscape stores bookmarks as HTML, Safari uses XML, and Opera keeps a simple ASCII list of bookmarks with their corresponding folders preceding them. Internet Explorer (IE) requires the user to export their bookmarks to the Netscape format because IE stores bookmarks in folders with one URL per file. Furthermore, Mozilla and Firefox both use the Netscape method of storing the bookmarks.

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 like descriptions that users entered, the time the bookmark was created and last accessed.

3.2 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. Then we combine all of the user trees into a single tree by

[1] Open Directory Project, `dmoz.org`

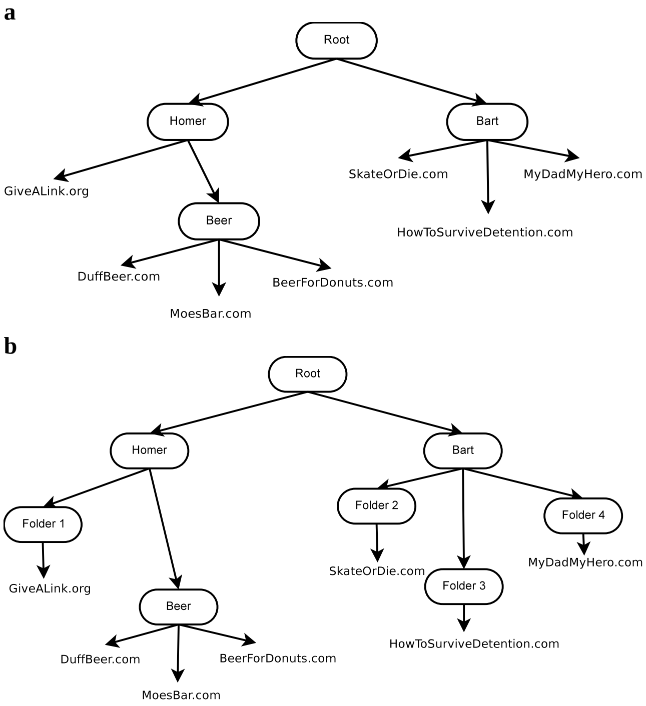


Figure 1: (a) An example tree containing the combined bookmark collection of two users, Homer and Bart. (b) The tree has been modified: each unclassified URL (i.e. each URL located in the user's root directory) has been given its own folder.

introducing a new root (super user) which is the parent of all user nodes. Figure 1(a) shows an example scenario in which only two users donated their bookmarks. Homer's bookmark collection contains `givealink.org` and also a folder called *Beer* that contains three URLs. Bart's collection is a flat file with three URLs. In our system, these URLs are stored as a single tree where the root is a parent of directories Homer and Bart, which are in turn the parents of the respective user's bookmarks.

To exploit the hierarchical structure of bookmark files, we use Lin's measure [8] to calculate similarity between the URLs in a user u's tree. Let URL x be in folder F^u_x, URL y be in folder F^u_y, and the lowest common ancestor of x and y be folder $F^u_{a(x,y)}$. 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 the root folder is $|U|$. Then the similarity between x and y according to user u is:

$$s_u(x,y) = \frac{2 \times \log\left(\frac{|F^u_{a(x,y)}|}{|U|}\right)}{\log\frac{|F^u_x|}{|U|} + \log\frac{|F^u_y|}{|U|}}. \qquad (1)$$

This function produces similarity values between 0 and 1. For example, if two URLs appear in the same folder, then their similarity is 1 because $F_x = F_y = F_{a(x,y)}$. Also, all other things being equal, the similarity between x and y is higher when F_y is a subfolder of F_x, than when F_x and F_y are siblings.

A downfall of this approach is that many Web users do not organize their bookmarks in folders and subfolders and instead keep a flat list with their favorite links. In this case, according to Lin's measure, all of the links are in the same folder, so they must have similarity 1 to each other, but in reality we cannot conclude strong semantic similarity between URLs listed in such unorganized files. Therefore if a user decided to leave some URLs in their root directory, we think of each URL as if it were in its own folder. Figure 1(b) depicts how we modify the tree from Figure 1(a) before calculating semantic similarity. In the modified structure, URLs listed at the root have similarity slightly higher than 0 (as opposed to 1 in the original structure).

Table 1: Protocols of bookmarks.

Protocol	Donations	%
http	19679	97.2 %
https	246	1.2 %
feed	123	1 %
file	97	< 1 %
ftp	53	< 1 %
javascript	33	< 1 %
gopher	1	< 1 %
other	13	< 1 %

According to Equation 1, two URLs donated by different users have $s_u = 0$ because the least common ancestor is the root (super user). Thus Lin's measure is only appropriate for calculating the similarity of URL pairs according to a single user. To calculate the global similarity between URLs x and y, we average the similarities reported by each user:

$$s(x,y) = \frac{1}{N}\sum_{u=1}^{N} s_u(x,y). \qquad (2)$$

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 calculate $s_u(x,y)$ for all locations of x and report the highest value. 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.

4. SIMILARITY NETWORK ANALYSIS

As of June 5, 2005, GiveALink has 113 users who have donated a total of 22,065 unique URLs. Based on this initial small number of donors, there is relatively little overlap of bookmarked URLs between different users. As a result the similarity matrix is very sparse. The protocols of the donated URLs are shown in Table 1.

Figure 2 shows the topology of the graph. The well defined clusters suggest that our semantic similarity measure is able to identify communities of Web pages that share a topic. One of the clusters, *News & Computers,* is particularly interesting because it reflects the interests of our early donors who are mostly Computer Science graduate students at Indiana University. It contains pages like the graduate students announcement board, the departmental site, and profiles from `thefacebook.com`.

To visually analyze the structure of the network we use *LaNet-vi*,[2] a layout algorithm based on k-core decomposition [2]. Like other visualization tools, *LaNet-vi* assumes an unweighted network and thus we must first use a threshold on the weights (similarities) to select edges. However, as shown in Figure 3, the similarity is distributed broadly, over three orders of magnitudes. This suggests that any threshold value on the similarity weights will lead to a loss of important structural information. Figure 3 also suggests that $s \approx 0.03$ is the critical region in which the weight distribution transitions into a power-law tail, and therefore this is the critical region where we expect to find interesting threshold values for the similarity. Therefore we visualize in Figure 4 three versions of the

[2] `xavier.informatics.indiana.edu/lanet-vi/`

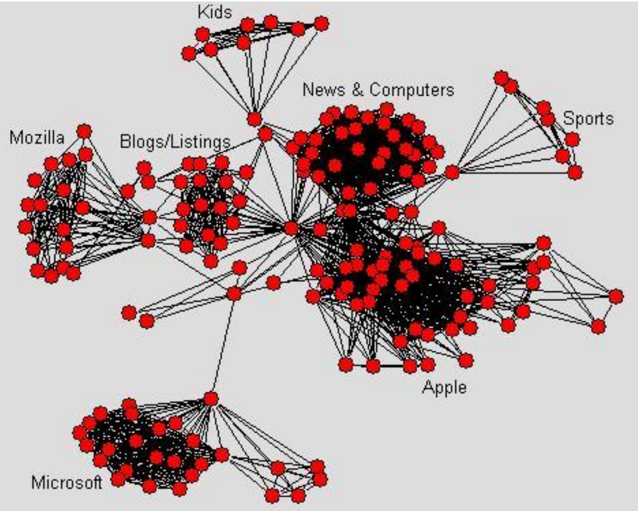


Figure 2: Graph topology generated using Pajek. Nodes displayed are those with at least two edges and edges with $s < 0.04$ have been removed. Labels are added by hand to reflect cluster content.

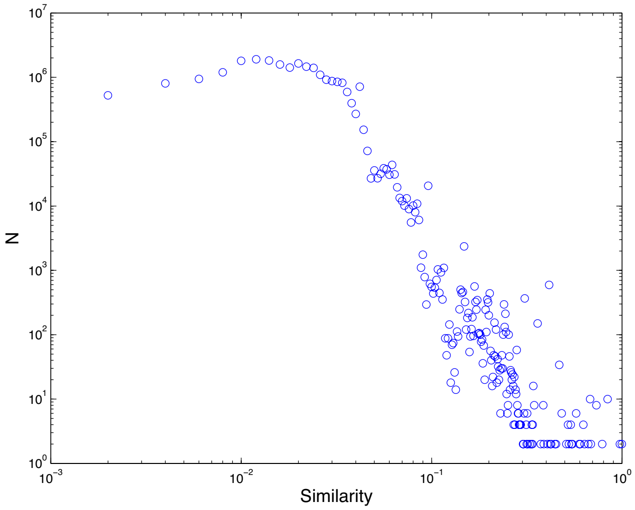


Figure 3: Distribution of the GiveALink link weights, i.e., the similarities $s > 0$ among all the pairs or URLs.

network corresponding to thresholds in this critical region. For high threshold values ($s > 0.04$) we can identify different clusters corresponding to those in Figure 2. As we consider more edges, this cluster structure is gradually lost as we gain more information on the topology. For $s > 0.02$ the network becomes very hierarchical and layered; the number of cores reaches 90. We also note that the degree of nodes appears to be strongly correlated with their centrality; intuitively general pages are similar to more pages than specific ones. Due to the sparsity of the similarity matrix and the distribution of s (cf. Figure 3), the majority of nodes appear isolated at these threshold values and are not displayed; we see at most 419 of the 22,065 URLs ($s > 0.02$).

Another way to analyze the GiveALink similarity network is to compare it with other ways to estimate semantic similarity between the bookmarked Web pages. In prior work [11, 9] we have compared similarity measures obtained from content (text) and hyperlink similarity, and a *semantic* similarity measure obtained from the ODP. As mentioned in § 2, the correlations between these measures across all pairs of pages in the ODP are quite low. To compare the GiveALink similarity, we focus on the intersection between GiveALink and the ODP, i.e., we retain the 1,496 bookmarked pages that are also in the ODP. This yields the correlations shown in Table 2. Since the semantic similarity is based on the golden standard of manual classification and validated by user assessments [9], we can use it as a reference. From this perspective, looking at the third column in Table 2, we find that GiveAlink is a worse predictor than text similarity, but a better predictor than link similarity. However all correlations are very low, suggesting that the collaborative filtering relationship in the GiveALink similarity yields yet a different kind of information than the other measures.

Table 2: Spearman correlation coefficients between different similarity measures. All differences are statistically significant.

	Link	GiveALink	Semantic
Content	0.055	0.045	0.138
Link		0.026	0.040
GiveALink			0.082

Figure 5 reinforces this view, showing that the topologies of the four similarity networks are qualitatively different. The content-based network is more regular, with correlated degree and centrality. The link-based and semantic networks are less regular and correlated. The GiveALink network has intermediate regularity and correlation between degree and centrality, but is very layered with 92 cores. Again we conclude that the different similarities provide us with different information about relationships among pages.

5. APPLICATIONS

5.1 Search

The pivotal application of the GiveALink project is a search system that allows users to explore the bookmark collection. Figure 6 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 the query, according to our similarity matrix s. Search results can be ranked according to a combination of four different measures: bookmark similarity and three additional ranking measures described below. If the user picks several ranking measures, then results are ranked by the product of their values. If the user does not pick any ranking measure, results are ranked by bookmark similarity to the query.

The GiveALink database is currently quite small and it is often the case that it will not contain the query URL provided by the user. If we do not have the query URL, it is impossible to estimate similarity between it and other URLs in the collection based on our similarity measures. In this case we resort to help from a search engine: 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 database and expand the resulting set with additional sites from our database similar to them. Finally, we rank the results in the same way as before, using a combination of similarity and ranking measures. Note that we only return URLs that are in our database, and therefore the similarity and ranking values are known for all of them. The additional URLs from our database carry similarity and ranking values with respect to the Google result that generated them.

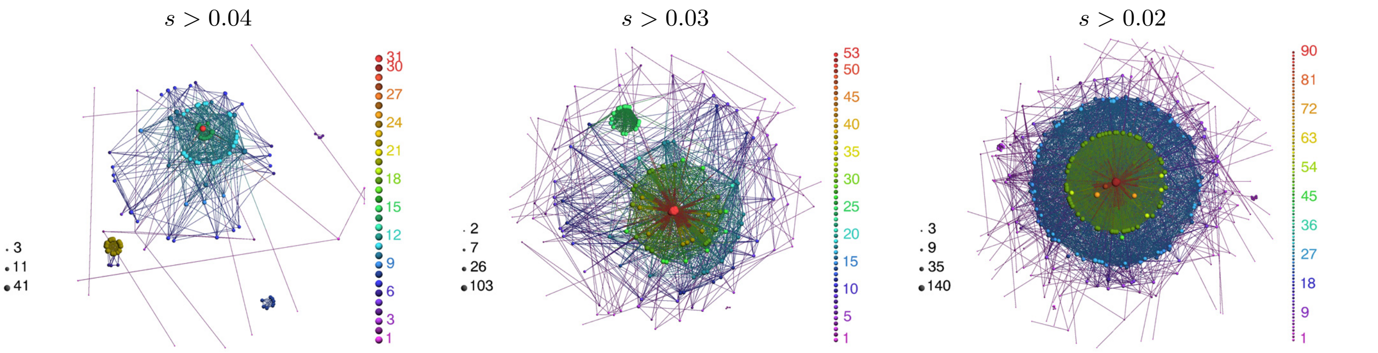


Figure 4: Visualizations of the GiveALink similarity network with different thresholds on edge weights. The *LaNet-vi* tool uses the size of a node to represent its degree (left legend) and colors to represent cores (right legend). Higher-numbered cores are the more internal components of the networks.

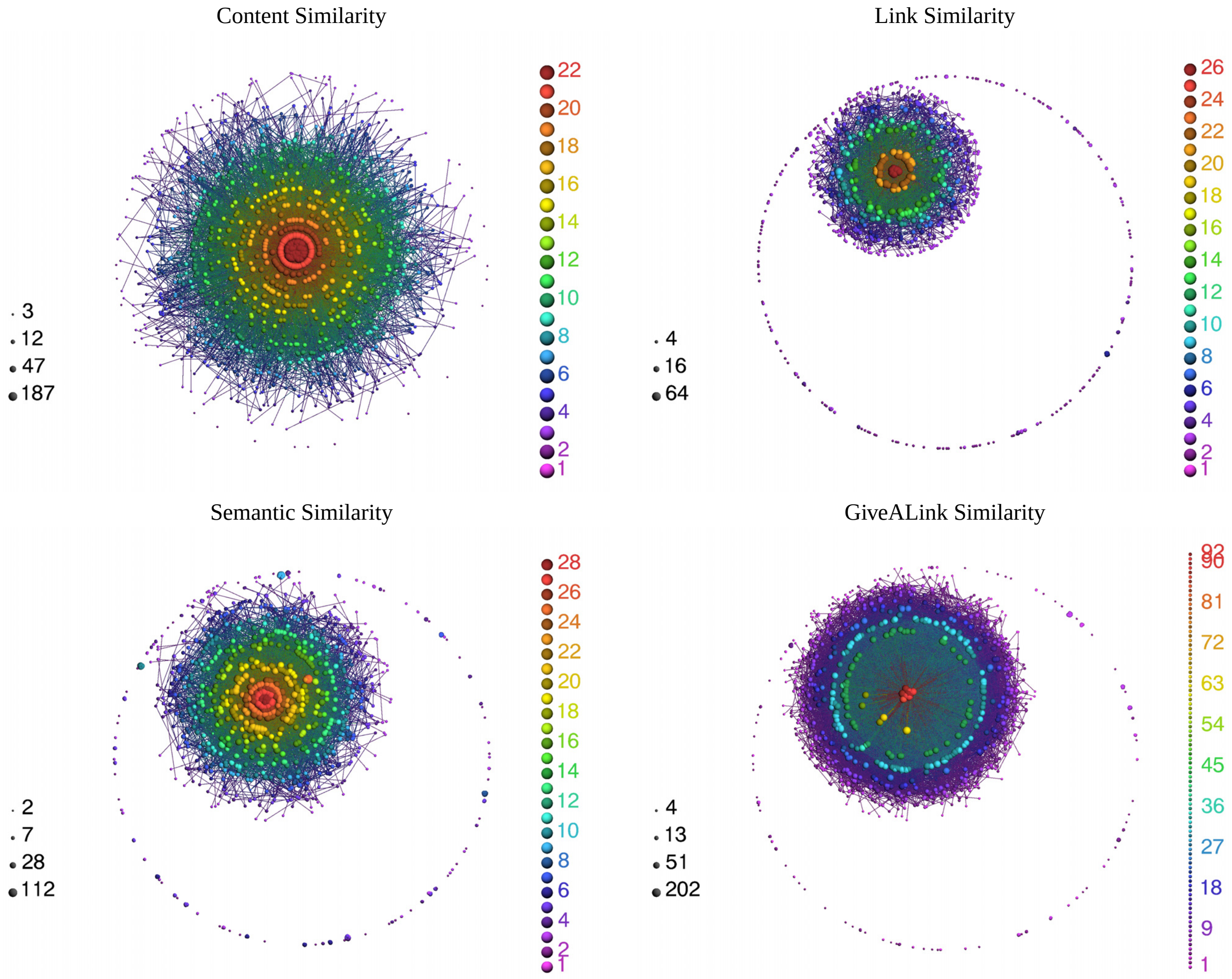
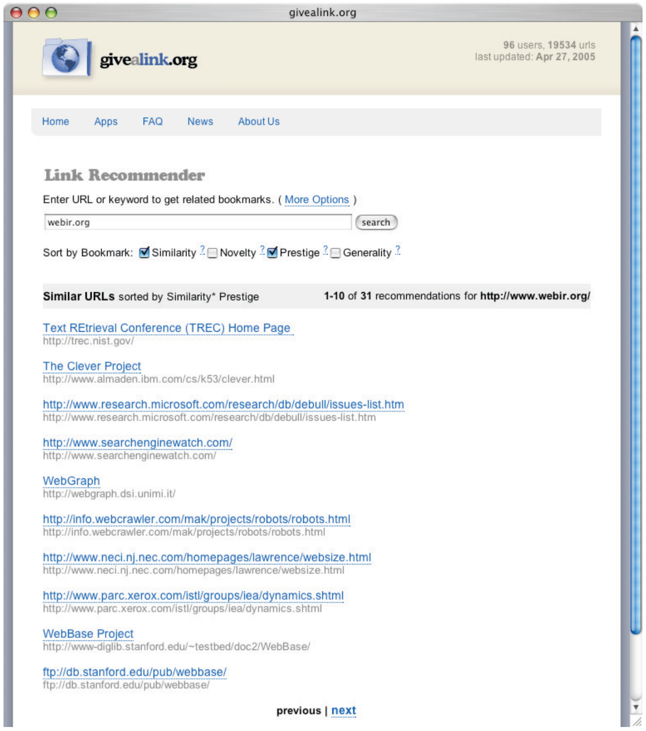


Figure 5: Visualizations of four similarity networks using the 1,496 pages that appear in both GiveALink and the ODP. For each similarity, a threshold is chosen to maximize the size of the network that can be visualized with the public *LaNet-vi* tool.



givealink.org

96 users, **19534** urls
last updated: **Apr 27, 2005**

Link Recommender

Enter URL or keyword to get related bookmarks. (More Options)

webir.org

Sort by Bookmark: ☑ Similarity ? □ Novelty ? ☑ Prestige ? □ Generality ?

Similar URLs sorted by Similarity* Prestige **1-10** of **31** recommendations for **http://www.webir.org/**

Text REtrieval Conference (TREC) Home Page
http://trec.nist.gov/

The Clever Project
http://www.almaden.ibm.com/cs/k53/clever.html

http://www.research.microsoft.com/research/db/debull/issues-list.htm
http://www.research.microsoft.com/research/db/debull/issues-list.htm

http://www.searchenginewatch.com/
http://www.searchenginewatch.com/

WebGraph
http://webgraph.dsi.unimi.it/

http://info.webcrawler.com/mak/projects/robots/robots.html
http://info.webcrawler.com/mak/projects/robots/robots.html

http://www.neci.nj.nec.com/homepages/lawrence/websize.html
http://www.neci.nj.nec.com/homepages/lawrence/websize.html

http://www.parc.xerox.com/istl/groups/iea/dynamics.shtml
http://www.parc.xerox.com/istl/groups/iea/dynamics.shtml

WebBase Project
http://www-diglib.stanford.edu/~testbed/doc2/WebBase/

ftp://db.stanford.edu/pub/webbase/
ftp://db.stanford.edu/pub/webbase/

Figure 6: Screen shot of the GiveALink search system.

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 processed in the way described above. As our bookmark collection grows, our goal is to make this system independent of external search engines. We plan to match the query keywords against the descriptions and titles that users enter for the URLs 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.

The similarity matrix described above provides one way of ranking search results: according to bookmark similarity s to the query URL. We also derive three other ranking measures that we refer to as *generality*, *prestige*, and *novelty*. They provide more ways to rank query results by taking into account aspects of the global associative semantics of the bookmark network. Generality and prestige provide total orders for the URLs in our collection that are independent of the user query. Novelty combines bookmark similarity of search results to the user query with an aspect of the global network, namely semi-metric distances.

5.1.1 *Prestige*

Prestige is a recursive measure inspired by Google's PageRank[3] — the prestige of a URL is tied to the prestige of its neighbors in the similarity graph. The difference between our prestige and PageRank is that PageRank is computed on a directed, unweighted graph where edges represent hyperlinks. Prestige is computed on an undirected, weighted graph in which the weights of edges represent social similarity s as defined in our similarity matrix. The iterative process is defined as follows: at time step $t = 1$, 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

$$P_i(t+1) = (1-\alpha) + \alpha \times \sum_j \frac{s(i,j) \times P_j(t)}{\sum_k s(j,k)}. \tag{3}$$

The computation continues until the prestige values converge. We use $\alpha = 0.85$.

5.1.2 *Generality*

Generality is the term selected to describe to our non-technical users the *centrality* of a URL node in the similarity matrix. 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.

Calculating centrality requires the computation of the similarity between all pairs of nodes according to all of the paths between them. There are many ways in which this can be done. 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 $x \rightsquigarrow y$ that goes through z, 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 too aggressive for computing similarities [14].

In our system, we convert similarity values to distances, then we compute shortest-path distances using Floyd-Warshall's algorithm [4], and finally we convert these values back into shortest-path similarity values. To convert between similarity and distance values, we use the following formula:

$$dist(x,y) = \frac{1}{s(x,y)} - 1. \tag{4}$$

Note that when similarity is equal to 1, distance is equal to 0, and when similarity is equal to 0, distance is infinity. Thus the closer two URLs are, the higher their similarity is. 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

$$s_{max}(x,y) = \left[1 + \min_{x \rightsquigarrow y} \sum_{(u,v) \in x \rightsquigarrow y} \left(\frac{1}{s(u,v)} - 1 \right) \right]^{-1}. \tag{5}$$

5.1.3 *Novelty*

A natural observation to make, once we have computed the all-pairs shortest-path similarities, is that 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 and 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 phenomenon, known as semi-metric behavior [13], is very valuable for a recommendation system because it reveals similarity that is implied by the global associative semantics but has not yet been discovered by individual bookmark users. If used in addition to a direct similarity measure, it empowers the recommendation system to not only generate recommendations that are natural and obvious, but also ones that are unexpected and could inspire users to broaden and deepen their interests.

We attempt to exploit semi-metric behaviors by a novelty measure defined as

$$novelty(x,y) = \begin{cases} \frac{s_{max}(x,y)}{s(x,y)} & \text{if } s(x,y) > 0 \\ \frac{s_{max}(x,y)}{s_{min}} & \text{if } s(x,y) = 0 \end{cases} \tag{6}$$

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 [13]. 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. As the gap grows, the novelty value grows as well.

We call this measure novelty because, when a user submits query x and our search engine returns answer y, where $s(x, y)$ is low but $s_{max}(x, y)$ is high, then y is a valid recommendation and is novel with respect to the measured associations of any one user. Indeed, the indirect associations captured by the novelty ratio are a global property of the network and cannot be locally measured from direct association levels [13]. If the user chooses to rank search results by novelty (or some combination of measures that include novelty), then the recommendations that are non-trivial and unexpected will be ranked higher.

5.2 Recommendation

A natural extension of ranking search results by novelty is a recommendation system that is roughly equivalent to searching by novelty. In the standard search system, results are generated by mining the database for URLs that have high bookmark similarity to the user query. Thus the standard search system is essentially "search by similarity." On the other hand, in the recommendation system, the results are URLs that have high *novelty* to the user query. Results are generated from different sets of URLs in the two applications. The search system considers all URLs in the database and picks the ones most similar to the query. The recommendation system only considers the URLs that have higher shortest-path similarity than direct similarity to the query, and picks the ones with highest novelty.

The two types of systems, search and recommendation, address different information needs. The search system provides additional information that is highly relevant to the user query; if the user provides a URL as the query term, the search results will perhaps expand on the knowledge already contained in the query URL. The recommendation system, on the other hand, provides different information that relates to the query in a non-trivial way. Rather than presenting similar information, recommendation results will provide pages that address the same questions from a different perspective: a different domain of knowledge, perhaps a different time period or geographical location. Thus the recommendation system could inspire collaboration between groups of users who do not yet realize they have similar interests.

5.3 Case Study

In Table 3 we illustrate the different ranking measures using the results of searching for `apple.com`. For comparison, we also present the pages that Google deems "similar," by submitting a `related:apple.com` query to Google. Google does not disclose how similar pages are identified[3] but it is safe to assume that a combination of text and link analysis is employed. As with the clustering seen in 2, these results are highly influenced by the interests of the system's early users as exemplified by the appearance of the IU computer science graduate student Web-board in the top ten results of prestige.

The results are quite exciting with regard to the obvious differences between rankings. Whereas nine of the ten results Google provides are corporate homepages, our Similarity ranking clearly values sites of practical interest to Mac users. Novelty appears to work as intended, revealing potentially relevant sites not listed among the other rankings.

6. CONCLUSIONS

In summary, we presented GiveALink, 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 but is qualitatively different, and weakly correlated with, other Web page networks built from similarity measures extracted from content, links, and classification ontologies. We also introduced a search and recommendation system with prestige, generality, and novelty ranking measures extracted from the similarity data.

An obvious advantage of our system when compared to traditional search engines 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, however, does not depend on the content of the page at all and we can recommend movie feeds, javascripts, URLs containing images only, files in various formats, and so on without having the means to access their contents.

Regarding coverage, we note that not all the URLs in our collection are known to Google. For example Google crawled relatively few of the HTTPS pages that people donated as bookmarks; this might be due to Robot Exclusion policies. In addition, 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.

Coverage is also the main current challenge for GiveALink. The sparsity of the bookmark similarity network is due to a small number of donors. We will attempt to achieve critical mass by removing barriers and creating incentives for users to donate bookmarks. For example, users must be convinced that their privacy will be honored by the system, and their use of GiveALink can be facilitated by Web services for search and recommendation.

Having looked at item-to-item (bookmark-to-bookmark) collaborative filtering using bookmark files, an obvious next step is person-to-person collaborative filtering. To this end there are two applications on which we plan to focus our future efforts:

Profile Based Recommendations This system will recommend sites based on the bookmarks that a particular user has submitted. Such recommendations are offered without a query being specified by the user. The process is such that users are clustered, and the most common or interesting URLs not found in a particular user's collection, but found among related users collections, are offered as recommendations.

Personalized Search Personalized search, like user profile recommendations, will be based on bookmarks submitted by an individual and the clustering of like individuals. Unlike user profile recommendations, the results are query-dependent. The results are a subset of "regular" search results that are relevant to the specific cluster of users to which the current user belongs. Users might also have multiple profiles based on subsets of their bookmark collection.

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

[3] `www.google.com/help/features.html#related`

Table 3: Recommendations for `apple.com`.

Similarity	Novelty	Prestige*Similarity	Generality*Similarity	Google (related:apple.com)
Apple–Store	Shockwave.com	CNN	Apple–Support	Apple
Apple–Support	CNET	Apple–Support	Apple–Store	Microsoft
Apple–SW Updates	Warehouse.com/Apple	Apple–Store	MSN	Adobe
Mactopia	Inside Mac Games	Google News	Mactopia	Sun
Office for Mac	MacWindows	MSN	Office for Mac	Macromedia
Apple Livepage	Apple–Product Guide	BBC News	Apple–SW Updates	HP
MSN	MozillaZine	Mapquest	Apple–Products	IBM
Apple–Products	MacToday	Apple–Quicktime	Apple–Hot News	Dell
Apple–Hot News	MacGamer	IU CS Webboard	Apple Livepage	Apple–MacOS X
Apple–MacOS X	MacGaming	Mactopia	Apple–MacOS X	Netscape

Acknowledgments

We would like to thank Luis Rocha for suggesting the novelty ranking measure and the recommendation system; Alessandro Vespignani for discussing the evaluation of the networks; Bill English for designing the GiveALink Web site; Rob Henderson and Jacob Ratkiewicz for invaluable technical help; and the NaN group at Indiana University for helpful discussions throughout the development of the project. This work was funded in part by NSF Career Grant IIS-0348940 to FM.

7. REFERENCES

[1] David Abrams, Ronald Baecker, and Mark H. Chignell. Information archiving with bookmarks: Personal web space construction and organization. In *CHI*, pages 41–48, 1998.

[2] Ignacio Alvarez-Hamelin, Luca Dall'Asta, Alain Barrat, and Alessandro Vespignani. k-core decomposition: A tool for the visualization of large scale networks. Computing Research Repository (CoRR), http://arxiv.org/abs/cs.NI/0504107, 2005.

[3] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[4] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorightms, 2nd ed.* MIT Press, 2001.

[5] Prasanna Ganesan, Hector Garcia-Molina, and Jennifer Widom. Exploiting hierarchical domain structure to compute similarity. In *ACM Trans. Inf. Syst. 21(1)*, pages 64–93, 2003.

[6] Tony Hammond, Timo Hannay, Ben Lund, and Joanna Scott. Social bookmarking tools (I): A general review. *D-Lib Magazine*, 11(4):doi:10.1045/april2005–hammond, 2005.

[7] Markus Jakobsson and Filippo Menczer. Web forms and untraceable ddos attacks. In S. Huang, D. MacCallum, and D.Z. Du, editors, *Network Security*, Forthcoming in June 2005.

[8] Dekang Lin. An information-theoretic definition of similarity. In *Proc. 15th International Conf. on Machine Learning*, pages 296–304. Morgan Kaufmann, San Francisco, CA, 1998.

[9] Ana Maguitman, Filippo Menczer, Heather Roinestad, and Alessandro Vespignani. Algorithmic detection of semantic similarity. In *Proc. WWW2005*, 2005.

[10] Bruce McKenzie and Andy Cockburn. An empirical analysis of web page revisitation. In *Proc. of the 34th Hawaii International Conference on System Sciences*, 2001.

[11] Filippo Menczer. Mapping the semantics of web text and links. *IEEE Internet Computing*, 9(3):27–36, May/June 2005.

[12] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstorm, and John Riedl. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work*, pages 175–186, Chapel Hill, North Carolina, 1994. ACM.

[13] Luis M. Rocha. Semi-metric behavior in document networks and its application to recommendation systems. In V. Loia, editor, *Soft Computing Agents: A New Perspective for Dynamic Information Systems*, pages 137–163. International Series Frontiers in Artificial Intelligence and Applications. IOS Press, 2002.

[14] Luis M. Rocha. Personal communication, 2005.

[15] Badrul M. Sarwar, George Karypis, Joseph Konstan, and John Riedl. Analysis of recommender algorithms for e-commerce. In *Proceedings of the 2nd ACM E-Commerce Conference (EC'00)*, 2000.

[16] Upendra Shardanand and Patti Maes. Social information filtering: Algorithms for automating "word of mouth". In *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, volume 1, pages 210–217, 1995.

[17] Brent Smith, Greg Linden, and Jeremy York. Amazon.com recommendations: item-to-item collaborative filtering. In *Internet Computing, IEEE*, 2003.

[18] Luis von Ahn, Manuel Blum, Nicholas J. Hopper, and John Langford. Captcha: Using hard ai problems for security. *Advances in Cryptology: Eurocrypt*, 2003.