Implicit Tagging using Donated Bookmarks

Ben Markines
bmarkine@cs.indiana.edu

Lubomira Stoilova
lstoilov@cs.indiana.edu

Filippo Menczer
fil@indiana.edu

Department of Computer Science, School of Informatics
Indiana University
Bloomington, IN 47406, USA

ABSTRACT

GiveALink measures a social semantic similarity between URLs by combining social collaboration with the hierarchical structure of bookmark files. The majority of other social bookmarking tools today require users to manually tag the URLs they submit. The tagging approach has some limitations including the need for users to specify labels, the ambiguity of tag names, and a lack of clear relationships between them. GiveALink explores a different approach by inferring a similarity from the folder organization of bookmark files. This approach is automated, and builds a global similarity network of all URLs. Relationships are made independently of the Web resource content, hence GiveALink is able to relate movies, images, and pages without needing to crawl their contents. Applications for search, recommendation, and personalization are made available exploiting the similarity relationship.

1. INTRODUCTION

GiveALink [6] is a site where users may submit and manage their bookmarks securely online. Donations are processed collectively, through collaborative filtering techniques in order to determine semantic similarity between URLs. Similarity is measured from the bookmark file structure where folders are used as implicit tags.

The majority of current social bookmarking tools require users to explicitly describe their URLs with labels or tags. This approach has some additional limitations.

Synonymous tags add a level of complexity for many tagging systems. Different labels with the same meaning may be applied to the same set of URLs. This may force users looking for similar links to potentially search in numerous tags and categories. An analogous difficulty stems from polysemy, i.e. a tag with multiple, different meanings.

The flat structure inherent in tagging also limits the ability to define relationships easily recognized in hierarchies. Consider for instance `en.wikipedia.org/wiki/Computer_program` labeled with `ComputerProgramming`, while `java.sun.com` may be tagged with `Java` and `ComputerProgramming`. This relationship may be better represented with the label `Java` as a child of `ComputerProgramming`. In this example, `java.sun.com` is considered to be more specific while preserving a strong similarity with `en.wikipedia.org/wiki/Computer_program` by being in the same subtree.

GiveALink goes beyond the simple tagging functionality by actively exploiting collaborative filtering and the hierarchical structure of donated bookmark files.

Synonymous and polysemous tags are not an issue in GiveALink. In fact tags are not used when determining similarity. The relationship between each link pair is computed. Hence we obtain a direct relationship between every pair of URLs without the need to navigate intermediate tags.

The hierarchy inherent in bookmark files is key for organizing URLs in GiveALink. Folder organization is an implicit way to tag and thus relationships between URLs are determined by their location in the bookmark file rather than the tag names. We use an established measure from information theory [2] to utilize the structure of bookmark files and to determine URL similarity. These values are then accumulated across all donations which collaborate to produce a large URL by URL similarity matrix. The more donations that organize resource x in proximity of resource y in the hierarchy, the better the similarity score. This technique requires less user participation (i.e. applying labels to URLs) because relationships are detected automatically in the bookmark files maintained by donors.

No special consideration is necessary for classifying multimedia in GiveALink. Because our system relies only on valid URLs, one only needs to place the URL in a common folder in order to establish similarity. Classifying an MP3 and a JPEG of your favorite artist works the same as placing CNN with ABC News in a folder.

Although we describe a technique utilizing the structure in bookmark files, it is not an absolute requirement for users to organize their bookmarks hierarchically. URL similarity is computed based on the collaborative filtering across all donations, making all bookmark files useful irrespective of internal structure.

2. DEMO

Users may donate at `givealink.org` either anonymously or as registered users. Anonymous donations are intended for users who wish to participate, but who do not wish to reveal their email address to the GiveALink site. Registered donors allow GiveALink to personalize their experience by building a profile for each user.

Registered users have full access to all of the personalized applications. Each registered participant has a personal home page that acts like a hub to GiveALink's two major personalized applications: personal recommendation and online bookmark management. Further details on these applications are given in Section 4.

The following public applications are also available with

WWW2006, May 22–26, 2006, Edinburgh, UK.
.

details in Section 4.

1. Search works similarly to popular search engines. Users may search either by URL or keywords and rank the results with three measures: similarity, novelty, and prestige. These measures may be selected individually or in combination. Details of these ranking measures are described by Stoilova *et al.* [6, 3].

2. The recommendation system generates novel links that are related in unconventional ways. Users may receive recommendations by URL or keywords. The set of URLs for recommendation are different from those in the search system. Details are explained by Stoilova *et al.* [6, 3].

3. Clients may subscribe to GiveALink's RSS for the latest URL rankings.

4. The similarity matrix is freely available for download at `givealink.org`.

5. Users may donate individual links through a bookmarklet.

3. SYSTEM DESIGN

To prevent spammer bots from polluting the database with engineered bookmark files, we require users to pass a CAPTCHA test when donating anonymously. In addition, we prevent multiple submissions of identical files (like default bookmark files) by checking the MD5 signature.

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 must send us an email with their activation code in the subject. We use relay information from the email to verify the source. This registration process is proposed by Jakobsson and Menczer [1] as an alternative to the double-opt in protocol and avoids email cluster bomb DDoS attacks.

When users donate at `givealink.org`, we parse their files by determining browser and platform from the user-agent header. Our set of parsers supports Internet Explorer, Netscape, Mozilla, Firefox, Safari, and Opera. Safari uses XML, but the latest version stores bookmarks in binary format. We developed a Web service[1] that converts Safari's binary format to ASCII.

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.

4. APPLICATIONS

4.1 Search

The pivotal application of the GiveALink project is a search system that allows users to explore the bookmark collection. When the user provides a query URL, the system looks for other URLs that have high bookmark similarity to it, according to our similarity matrix. Search results can be ranked according to a combination of three different measures: bookmark similarity and two additional ranking measures described by Stoilova *et al.* [6, 3]. 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 similarity to the query. 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 used to expand the resulting set with additional sites from our database similar to them.

[1] `homer.informatics.indiana.edu/cgi-bin/plutil/plutil.cgi`

4.2 Recommendation

The recommendation system generates results that are novel and related to the search query in non-obvious ways. In particular, the system looks for URLs where the indirect similarity to the query URL (computed by going through a third URL) is higher than the direct similarity. For example in the Simpsons, Moe and Bart are not very similar to each other directly, but they are indirectly related because both of them spend a lot of time with Homer. Similarly `amazon.com` and `netflix.com` do not have a high direct similarity in our system, so the search system does not return one when the user searches for the other. The recommendation system however recognizes that both of them are similar to `imdb.com` in that they publish movie reviews, so `amazon.com` is a valid recommendation when a user searches for URLs related to `netflix.com`.

4.3 Personalized Search

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. The measure that we use is the maximum similarity between the given URL and a URL in the user's bookmark collection.

In addition, when personalizing search results, we would like to pay particular attention to the interests of the user that are unique and stand out in respect to the other users. Thus we weigh the similarity between a URL and a user's bookmark by how unlikely it is that the user has that bookmark, in a way analogous to the inverse document frequency in the TFIDF [5] weighting scheme.

When a user submits a search query to the personalizing engine, we retrieve search results in the same way as for the general search engine. The personalized similarity score is then treated as another ranking measure: the search results are ranked by the product of their similarity to the query, times their similarity to the user profile, times the values of any other ranking measures the user selected.

4.4 Other Services

Some additional applications are part of the system to make our data more accessible: an RSS feed, a bookmark management application, and a bookmarklet.

The RSS feed returns GiveALink's search results in XML format. Users may either treat this as a Web service or as a channel for related URLs and queries that can be ordered using the different ranking measures. Also available through the feed are the most prestigious URLs.

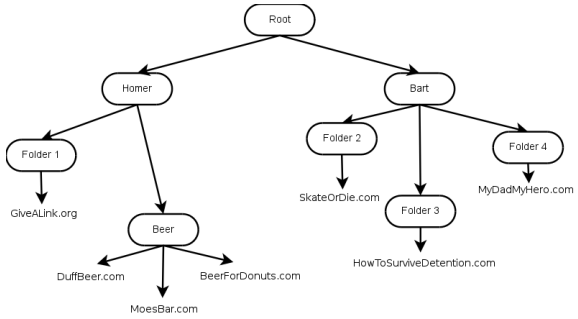

Figure 1: A tree from two users' bookmarks. Flat bookmark files are automatically structured with folders to extract meaningful similarity information.

The bookmark management application is intended to become an interface for users to manage their bookmarks, and to encourage them to organize their links in their personal Web directory. Currently users may insert, delete, move, copy, and share their collections.

The bookmarklet allows users to donate individual links. At this time, links are added to the top level of a user's tree.

5. MEASURING 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 introducing a new root (super user) which is the parent of all user nodes. Figure 1 shows an example scenario in which only two users donated their bookmarks.

To exploit the structure of bookmark files, we use Lin's [2] 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 the root folder is $|R|$. Then the similarity between x and y according to user u is:

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

This function produces similarity values in $[0,1]$. For example, if two URLs appear in the same folder, 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.

Many Web users do not organize their bookmarks in folders and subfolders and instead keep a flat list with their favorite links. If a user decides to leave some URLs in the root directory, we think of each URL as if it were in its own folder.

According to Equation 1, two URLs donated by different users have $s_u = 0$ because the lowest 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 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 calculate $s_u(x,y)$ for all locations of x and 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 share x and y, the higher $s_u(x,y)$. 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. Note that for users whose bookmarks are unorganized (flat), the similarity measure reverts to standard collaborative filtering [4].

6. CONCLUSION

GiveALink utilizes the structure of bookmark files in place of explicit online tagging. When a user donates their bookmarks, the system automatically detects similarities with no further user manipulation. In the absence of hierarchical structure in the bookmarks, the system resorts to collaborative filtering.

In order to improve the scalability of the system, we are exploring an adaptive thresholding mechanism to preserve the sparsity of the similarity matrix. To this end we exploit the modularity of the matrix. If one observes each user as having their own sub-matrix, then $s_u(x,y)$ is independent of $s_{u'}(x',y')$ in *most* cases where u and u' are different contributors with distinct URLs x, y, x', y'.

We conducted a user study [3] comparing ranking criteria from GiveALink and Google's "related" service. Each subject submited query URLs and determined whether each resulting URL was related to it. From the data collected, precision and recall were calculated and averaged across all queries. GiveALink performed competitively with Google. We are encouraged by this result and confident that our system will improve with more participation from Web users.

7. REFERENCES

[1] M. Jakobsson and F. Menczer. Web forms and untraceable DDoS attacks. In S. Huang, D. MacCallum, and D. Du, editors, *Network Security*, 2005.

[2] D. Lin. An information-theoretic definition of similarity. In *Proc. 15th Intl. Conf. on Machine Learning*, pages 296–304. Morgan Kaufmann, 1998.

[3] B. Markines, L. Stoilova, and F. Menczer. Bookmark hierarchies and collaborative recommendation. In *Proc. AAAI Conf.*, 2006. Forthcoming.

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

[5] K. Sparck-Jones, S. Walker, and S. Robertson. A probabilistic model of information retrieval: Development and comparative experiment. In *Information Processing and Management*, 36(1-2):1, pages 197-206, and 2, pages 809-840, 2000.

[6] L. Stoilova, T. Holloway, B. Markines, A. Maguitman, and F. Menczer. GiveALink: Mining a Semantic Network of Bookmarks for Web Search and Recommendation. In *Proc. KDD Workshop on Link Discovery: Issues, Approaches and Applications*, 2005.