

# HiBO: Mining Web's Favorites

Sofia Stamou, Lefteris Kozanidis, Paraskevi Tzekou, Nikos Zotos,  
and Dimitris Cristodoulakis

Computer Engineering and Informatics Department, Patras University,  
26500 Patras, Greece  
{stamou, kozanid, tzekou, zotosn, dxri}@ceid.upatras.gr

**Abstract.** HiBO is a bookmark management system that incorporates a number of Web mining techniques and offers new ways to search, browse, organize and share Web data. One of the most challenging features that HiBO incorporates is the automated hierarchical structuring of bookmarks that are shared across users. One way to go about organizing shared files is to use one of the existing collaborative filtering techniques, identify the common patterns in the user preferences and organize bookmarked files accordingly. However, collaborative filtering suffers from some intrinsic limitations, the most critical of which is the complexity of the collaborative filtering algorithms that inevitably leads to the latency in updating the user profiles. In this paper, we address the dynamic maintenance of personalized views to shared files from a bookmark management system perspective and we study ways of assisting Web users share their information space with the community. To evaluate the contribution of HiBO, we applied our Web mining techniques to manage a large pool of bookmarked pages that are shared across community members. Results demonstrate that HiBO has a significant potential in assisting users organize and manage their shared data across web-based social networks.

**Keywords:** Hierarchical Structures, Web Data Management, Bookmarks, System Architecture, Personalization.

## 1 Introduction

Millions of people today access the plentiful Web content to locate information that is of interest to them. However, as the Web grows larger there is an increasing need in helping users to keep track of the interesting Web pages that they have visited so that they can get back to them later. One way to address this need is by maintaining personalized local URL repositories, widely known as bookmarks [15]. Bookmarks, also called *favorites* in the Internet Explorer, enable users to store the location (address) of a Web page so that they can revisit it in the future without the need of remembering the page's exact address. People use bookmarks for various reasons [1]: some bookmark URLs for fast access, others bookmark URLs with long names that they find hard to remember, yet others bookmark their favorite Web pages in order to share them with a community of users with similar interests.

As the number of the pages that are available on the Web keeps growing, so does the number of the pages stored in personal Web repositories. Moreover, although users visit frequently their bookmarked URLs, they rarely delete them; which practically results into users keeping stale links in their personal Web repositories. As a consequence, people tend to maintain large, and possibly overwhelming, bookmark collections [16]. However, keeping a flat list of bookmark URLs is insufficient for tracking down previously visited pages, especially if we are dealing with a long list of favorites. As the size of the personal repositories increases, the need for organizing and managing bookmarks becomes prevalent. To assist users organize their bookmark URLs in a meaningful and useful manner, there exist quite a few bookmark management systems offering a variety of functionalities to their users. These functionalities enable users to store their bookmarks into folders and subfolders named for the sites they are found in or named for the information they contain, as well as to organize the folders in a tree-like structure. Moreover, commercial bookmark management tools, e.g. *BlinkPro* [2], *Bookmark Tracker* [3], *Check and Get* [4], *iKeepBookmarks* [5], provide users with a broad range of advanced features like detection of duplicate bookmarks and/or dead links, importing, exporting and synchronizing bookmarks across different Web browsers (Mozilla, Internet Explorer, Opera, Netscape), updating bookmarks and so forth.

In this paper, we present HiBO; an intelligent system that automatically organizes bookmarks into a hierarchical structure. HiBO is a powerful bookmark management system that exploits a multitude of Web mining techniques and offers a wide range of advanced services. Most importantly, HiBO is a non-commercial research project for managing the proliferating data in peoples' personal Web repositories without any user effort. The main difference between HiBO and the other available bookmark management systems (cf. [11], [14], [15]) is that HiBO uses a built-in subject hierarchy for automatically organizing bookmarks within both the users' local and shared Web repositories. The only input that our approach requires is a hierarchy of topics that one would like to use and a list of bookmark URLs that one would like to organize into these topics. Through the exploitation of the hierarchy, HiBO delivers personalized views to the shared files and eventually it assists Web users share their information space with the community.

The remainder of the paper is organized as follows: we begin our discussion with the description of HiBO's architecture. In Section 3, we give a detailed description of the functionalities and services that our bookmark management system offers. Experimental results are presented in Section 4. We finally review related work and conclude the paper in Section 6.

## 2 Overview of HiBO Architecture

HiBO evolved in the framework of a large research project that aimed at the automatic construction of Web directories through the use of subject hierarchies. The subject hierarchy that HiBO uses contains a total of 475 topics organized into 14 top level topics, borrowed from the top categories of the Open Directory Project (ODP) [6]. At a high level, the way in which HiBO organizes bookmarks proceeds as follows: firstly HiBO downloads all the Web pages that have been bookmarked by a user

and process them one by one in order to identify the important terms inside every page. Important terms of a page are linked together formulating a lexical chain. Then, our system uses the subject hierarchy and the lexical chains to compute a suitable topic to assign to every page. Finally, HiBO sorts the Web pages organized into topics in terms of their relevance to the underlying topics. More specifically, given a URL (bookmark) HiBO performs a sequence of tasks as follows: (i) download the URL and parse the HTML page, (ii) segment the textual content of the page into shingles and extract the page's thematic words using the lexical chaining technique [8], (iii) map thematic words to the hierarchy's concepts and traverse the hierarchy's matching nodes upwards until reaching to one or more topic nodes, (iv) compute a relevance score of the page to each of the matching topics, (v) index the URL in the topic of the greatest relevance score. Figure 1 illustrates HiBO's architecture.

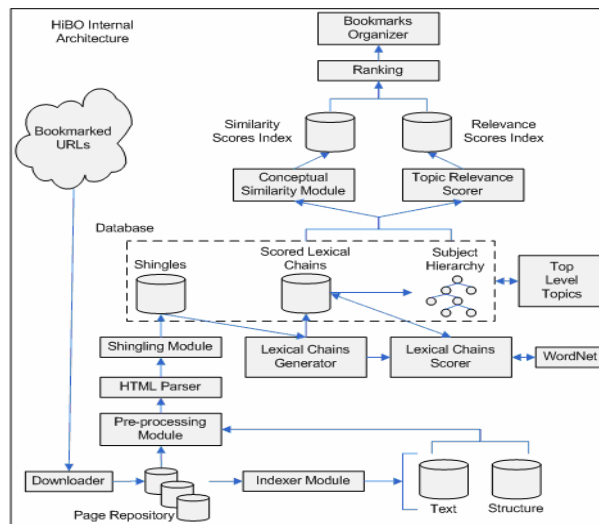


Fig. 1. Overview of HiBO architecture and functionality

In particular, after downloading and segmenting a Web page into shingles, HiBO generates a lexical chain for the page as follows: it selects a set of candidate terms from the page and for each candidate term it finds an appropriate chain relying on the type of links that are used in WordNet [7] for connecting the candidate term to the other terms that are already stored in existing lexical chains. If this is found, HiBO inserts the term in the chain and updates the latter accordingly. Lexical chains are then scored in terms of their elements' depth and similarity in WordNet, and their elements are mapped to the hierarchy's nodes. For each of the hierarchy's matching nodes, HiBO follows their hypernymy links until reaching a top level topic in which to categorize the Web page. Finally, HiBO sorts the Web pages categorized in each topic in terms of both the pages' conceptual similarity to one another and their relevance to the underlying topic. In estimating the pages' conceptual similarity, HiBO compares the elements in a page's lexical chain to the elements in the lexical chains of the other pages in the same topic, based on the assumption that the more elements the chains of

two pages have in common, the more correlated the pages are to each other. On the other hand, in computing the pages' relevance to the hierarchy's topics, HiBO relies on the pages' lexical chains scores and the fraction of the chains' elements that match a given topic in the hierarchy. Based on this general and open architecture, HiBO explores a variety of Web mining techniques and provides users with a number of advanced functionalities that are presented below.

### 3 HiBO Functionalities

**Organizing Bookmarks:** Besides the conventional way to organize bookmarks into a hierarchy of user-defined folders and subfolders, HiBO also incorporates a built-in subject hierarchy and a classification module, which automatically assigns every bookmarked page to a suitable topic in the hierarchy. HiBO's classification module is set into forth by the user and helps the latter structure her bookmarks in a meaningful yet manageable structure, instead of simply keeping a flat list of favorite URLs.

The subject hierarchy upon which HiBO currently operates is the one introduced in the work of [19]. Nevertheless HiBO's architecture is quite flexible to incorporate any hierarchy of topics that one would like to use. For automatically classifying bookmarks into the hierarchy's topics HiBO adopts the TODE classification technique, reported in [20]. At a very high level TODE classification scheme proceeds as follows: First, it processes the bookmarked pages one by one, identifies the most important terms inside every page and links them together, creating "lexical chains" [8]. Thereafter, it maps the lexical elements in every page's chain to the hierarchy's concepts and if a matching is found it traverses the hierarchy's nodes upwards until it reaches a top level topic. To accommodate for chain elements matching multiple hierarchy topics, TODE computes for every page a Relatedness Score (*RScore*) to each of the matching topics. *RScore* indicates the expressiveness of each of the hierarchy's topics in describing the bookmarked pages' contents. Formally, the relatedness score of a page  $p_i$  (represented by the lexical chain  $C_i$ ) to the hierarchy's topic  $T_k$  is determined by the fraction of words in the page's chain that are descendants (i.e. specializations) of  $T_k$ . Formally, the *RScore* of a page to each of the hierarchy's matching topics is given by:

$$RScore_k(p_i) = \frac{|\text{thematic words in } p_i \text{ matching } K|}{|\text{thematic words in } p_i|}. \quad (1)$$

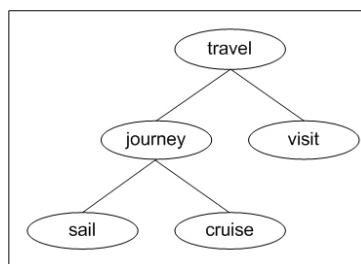
In the end, HiBO employs the topical category for which a bookmark has the highest relatedness score of all its *RScores* to describe that page's thematic content. By enabling bookmarks' automatic organization into a built-in hierarchical navigable structure, HiBO assists the user, who may be overwhelmed by the amount of her favorite pages organize and manage them instantly. Hierarchically organized bookmarks are stored locally on the user's site for future reference.

Moreover, HiBO supports personalized bookmarks' organization by enabling the user define the set of topics in which bookmarks would be organized. These topics can be either a subset of the hierarchy's topics or any other topic that the user decides. In case the user edits a new topic category in HiBO, she also needs to indicate a topic in HiBO's built-in hierarchy with which the newly inserted topic correlates. Through

the HiBO interface, the user can view the topics available in HiBO as well as the number of bookmarks in each topic. The user can navigate through the hierarchical tree to locate bookmarks related to specific topics.

In the case of shared bookmarks across a user community, HiBO supports personalized bookmark management by providing different views across users or user groups. Personalized views, allow the user decide on the classification scheme in which her shared bookmarks will be displayed. For instance, a user might choose to view the bookmarks she shares with a Web community organized in her self-selected categories or alternatively organized in the system's built-in subject hierarchy. Optionally, a user might decide to view her shared bookmarks organized in the categories defined by another member of the community, who she trusts. To enable personalized views on shared bookmarks, HiBO's classification module re-assigns user favorites to the categories preferred by the user (self, community or system defined) following the categorization process described above. Additionally, HiBO enables bookmark organization by their file types.

**Searching Bookmarks:** HiBO incorporates a powerful search mechanism that allows users to explore bookmark collections. The queries that HiBO supports are of the following types: topic-specific search, site/domain search, temporal search and keyword search. Similarly to querying a search engine for finding information on the Web, querying HiBO for locating information within one's Web favorites enables users to issue queries and retrieve bookmark URLs that are relevant to the respective queries. Upon keyword-based search, the user submits a natural language query and the system's search mechanism looks for bookmarked pages that contain any of the user-typed keywords, simply by employing traditional IR string-matching techniques. Additionally, HiBO incorporates a query refinement module introduced in the work of [12] and provides information seekers with alternative query formulations. Alternative query wordings are determined based on the semantic similarity that they exhibit to the user selected keywords in WordNet hierarchy. Refined queries are visualized in a graphical representation, as illustrated in Figure 2 and allow the user pick any of the system suggested terms either for reformulating a query that returns few or no relevant pages, or for crystallizing an under-specified information need.



**Fig. 2.** A refined query graph example

Moreover, HiBO supports topic-specific searches by allowing users select the topical category (e.g. folder) out of which they wish to retrieve search results. Topic-specific searches greatly resemble the process of querying particular categories in

Web Directories, in the sense that the user firstly selects among the topics offered in the HiBO hierarchy the one that is of interest to her and thereafter she issues and executes the query against the index of the selected topic. Search results can be ranked according to the query-bookmark similarity values combined with any of the measures described in the following paragraph. If the user selects multiple ranking measures, then results are ranked by the product of their values. Conversely, if the user does not pick a particular ranking measure, results are ranked by the semantic similarity between the query keywords (either organic, i.e. user typed, or refined, i.e., system suggested) to the terms appearing in the bookmark pages that match the respective query.

**Ranking Bookmarks:** HiBO provides several options for sorting the bookmarks listed in each of the hierarchy's topics as well as for sorting bookmarks that are retrieved in response to a user query. For ranking bookmark URLs that are retrieved in response to some query  $q$ , HiBO relies on the semantic similarity between the query itself and the bookmark pages that contain any of the query terms. To measure the semantic similarity between the terms in a query and the terms in the pages that match the given query, we use the similarity measure presented in [18], which is established on the hypothesis that the more information two concepts share in common, the more similar they are. The information shared by two concepts is indicated by the information content of their most specific common subsumer. Formally the semantic similarity between words,  $w_1$  and  $w_2$ , linked in WordNet via a relation  $r$  is given by:

$$\text{sim}_r(w_1, w_2) = -\log P(\text{mcs}(w_1, w_2)). \quad (2)$$

The measure of the most specific common subsumer (*mcs*) depends on: (i) the length of the shortest path from the root to the most specific common subsumer of  $w_1$  and  $w_2$  and (ii) the density of concepts on this path. Based on the semantic similarity values between the query terms and the terms in a page, we compute the average Query-Page similarity (QPsim) as:

$$\text{QPsim}(q(t), P(t)) = \frac{\sum_{p=1}^{P(t)} \text{sim}(q(t), P(t))}{|P(t)|}. \quad (3)$$

where  $q(t)$  denotes the terms in a query and  $P(t)$  denotes the terms in  $P$  that have some degree of similarity to the query terms. The greater the similarity value between the terms in a bookmark page and the terms in a query, the higher the ranking that the page will be given for that query.

On the other side of the spectrum, for ordering bookmarks in the hierarchy's topics, the default ranking that HiBO uses is the DirectoryRank (DR) metric [13], which determines the bookmarks' importance to particular topics as a combination of two factors: the bookmarks' relevance to their assigned topics and the semantic correlation that the bookmarks in the same topic exhibit to each other. In the DR scheme, a page's importance with respect to some topic is perceived as the amount of information that the page communicates about the topic. More precisely, to compute DR with respect to some topic  $T$ , we first compute the degree of the pages' relatedness to topic

$T$ . Formally, the relatedness score of a page  $p$  (represented by a set of thematic terms<sup>1</sup>) to a hierarchy's topic  $T$  is defined as the fraction of the page's thematic words that are specializations of the concept describing  $T$  in the HiBO hierarchy, as given by Equation (1). The semantic correlation between pages  $p_1$  and  $p_2$  is determined by the degree of overlap between their thematic words, i.e. the common thematic words in  $p_1$  and  $p_2$  as given by:

$$\text{Sim}(p_1, p_2) = \frac{2 \cdot |\text{common words}|}{|\text{words in } p_1| + |\text{words in } p_2|}. \quad (4)$$

DR defines the importance of a page in a topic to be the sum of its topic relatedness score and its overall correlation to the fraction of pages with which it correlates in the given topic. Formally, consider that page  $p_i$  is indexed in topic  $T_k$  with some  $\text{RScore}_k(i)$  and let  $p_1, p_2, \dots, p_n$  be pages in  $T_k$  with which  $p_i$  semantically correlates with scores of  $\text{Sim}(p_1, p_i), \text{Sim}(p_2, p_i), \dots, \text{Sim}(p_n, p_i)$ , respectively. Then the DR of  $p_i$  is given by:

$$\text{DR}_{T_k}(p_i) = \text{RScore}_k(i) + \frac{[\text{Sim}(p_1, p_i) + \text{Sim}(p_2, p_i) + \dots + \text{Sim}(p_n, p_i)]}{n}. \quad (5)$$

where  $n$  corresponds to the total number of pages in topic  $T_k$  with which  $p_i$  semantically correlates. Moreover, HiBO offers personalized bookmark sorting options such as the ordering of pages by their bookmark date or by their last update, as well as the ordering of bookmarks in terms of their popularity, where popularity is determined by the frequency with which a user or group of users sharing files, (re)visit bookmarks.

**Sharing Bookmarks:** Besides offering bookmark management services to individuals; HiBO constitutes a social bookmark network, as it allows community members share their Web favorites. In this perspective, HiBO operates as a bookmark recommendation system since it not only gathers and distributes individually collected URLs but it also organizes and processes them in a multi-faceted way. In particular, HiBO despite offering personalized views to shared bookmarks (cf. Organizing Bookmarks paragraph) it enables users annotate their preferred Web data, share their annotations with other members of the network and comment on others' annotations. To assist Web users exploit the knowledge accumulated in the bookmarks of others, HiBO goes beyond traditional collaborative filtering techniques and applies a multitude of Web mining techniques that exploit the hierarchical structure of the shared bookmarks. Such Web mining techniques range from the automatic classification of bookmark pages into a shared topical hierarchy, to the structuring of shared files according to their links and content similarity. Shared bookmarks' dynamic categorization is achieved through the utilization of the TODE categorization scheme, whereas bookmarks' structuring is supported by the different ranking algorithms that HiBO incorporates. Additionally, HiBO provides recommendation services to its users as it examines common patterns in the bookmarks of different community members and suggests interesting sites to users who might not have realized that they share common interests with others. HiBO communicates its recommendations in the form of

<sup>1</sup> The thematic terms in a page  $p$  are the lexical elements that formulate the lexical chain of  $p$ .

highlighted URLs that are associated to one's favorites, which are either stored in the system's hierarchy or retrieved in response to some query.

**Keeping Bookmarks Fresh:** Based on the observation that users rarely refresh their personal Web repositories, we equipped HiBO with a powerful update mechanism, which aims at maintaining the bookmarks index fresh. By fresh we mean that the index does not contain obsolete links among one's bookmarks, as well as that it reflects the current content of bookmarked pages. The update mechanism that HiBO uses performs a dual task: on the one hand it records the users' clickthrough data on their bookmarks and on the other it submits periodic requests to a built-in crawler for re-downloading the content of the bookmarked URLs. In case the system identifies bookmarks that have not been accessed for a long time, it posts a request to the user asking if she still wants to keep those bookmarks in her collection and/or if she still wants to share those bookmarks with other community members. Upon the user's negative answer, the system deletes those rarely visited URLs from the bookmark index and updates the latter accordingly, i.e. it re-orders pages etc. Similarly, if the system detects invalid, broken or obsolete URLs within a user's personal repository, it issues a notification to the user, who decides what to do with those links (either delete them, expunge them from her shared files, or keep them anyway). Furthermore, if the system detects a significant change in the current content of pages that had been bookmarked by a user some time ago, it issues an alert to the latter that her bookmarked URLs do not reflect the current content of their respective pages. It is then up to the user to decide whether she wants to keep the old or the new content of a bookmarked page. For content change detection, HiBO relies on the semantic similarity module discussed above, and uses a number of heuristics for deciding whether a page has significantly changed and therefore the user needs to be notified. HiBO's update mechanism although operates on a single user's site, it indirectly impacts the rest community members in the sense that upon changes in one's personal Web repository, these will be reflected on her shared files. Note that the update mechanism that HiBO embodies is optional to the user who might decide not to activate it and therefore not to be disturbed by the issued update alerts and notifications.

## 4 Experimental Setup

To evaluate HiBO's effectiveness in managing and organizing Web favorites, we launched a fully functional version of our bookmark management system and we contacted 25 postgraduate students from our school asking them to donate their bookmarks. Donating bookmarks pre-requisites that users register to the system by providing a valid e-mail address and they receive a personal code, which is used in all their transactions with the system. Upon code's receipt users obtain full rights on their personal bookmarks and they can also indicate the HiBO community with which they wish to share their preferred URLs. In the experiments reported here, all our 25 users formulated a single Web community sharing bookmarks. When users donate bookmarks, we use their agents to determine which browser and platform they are using in order to parse the files accordingly. We also use an SQL database server at the back-end of the system, where we store all the information handled by HiBO, i.e. users and



user groups, URLs, bookmarks' structure at the user site, the subject hierarchy, time stamps, clickthrough data, queries, etc. In our experiments, we used a total set of 3,299 bookmarks donated by our subjects and we evaluated HiBO's performance in automatically categorizing bookmarks in the system's hierarchy, by comparing its classification accuracy to the accuracy of a Bayesian classifier and a Support Vector Machine (SVM) classifier. We also investigated the effectiveness of HiBO's ranking mechanisms in offering personalized rankings. Table 1 summarizes some statistics on our experimental dataset.

**Table 1.** Statistics on the experimental dataset

# of bookmark URLs	3,299
# of users	25
# of topics considered	86
# of queries	48
Avg. # of bookmarks per user	131.96
Avg. # of shared bookmarks per user	58
Avg. # of topics per user	21
Avg. # of shared topics	9.4
Avg. # of queries per user	7.5
Avg. # of visited pages per query	5.8
Avg. # of useful pages per query	3.5
Avg. # of terms per refined query	3.8

To evaluate HiBO's efficiency in categorizing bookmarks to the hierarchy's topics, we picked a random set of 1,350 pages from our experimental data that span 18 topics in the Open Directory that are also among our hierarchy's topics and we applied our categorization scheme. Obtained results were compared to the results delivered by both the SVM and the Bayesian classifier that we trained with the 90% of the same dataset. Classification results are reported in Table 2, where we can see clearly that HiBO's classifier significantly outperforms both Bayesian and SVM classification with a notable performance; reaching to a 90.70% overall classification accuracy.

In Table 3, we illustrate the different ranking measures of HiBO, using the results of both browsing and searching for spam. For comparison, we also present the pages that Google considers "important" to the query spam. Although, Google uses a number of non-disclosed factors for computing the importance of a page, with PageRank [17] being at the core, we assume that a combination of content and link analysis is employed. Obtained results demonstrate the differences between the two HiBO rankings examined. In particular, the rankings delivered by DR sort bookmark pages in terms of their content importance to the underlying topic, i.e. Spam. As we can see from the reported data, our DR ranking values highly pages of practical interest compared to the pages retrieved from Google, which are general sites that mainly provide definitions of spam. On the other hand, the similarity ranking orders the bookmarked pages that are retrieved in response to the query spam in terms of their content semantic closeness to the semantics of the query. As such the results retrieved by HiBO contain pages that even if they are not categorized in the topic Spam, their contents exhibit substantial semantic similarity to the issued query. Recall that our experiments

were conducted towards a set of bookmarks that are shared across our subjects and as such reported results are influenced by our users' interests. This is exemplified by the appearance of Spam Filter for Outlook, Block Referrer Spam and Spam Fixer in the top ten results of DR and Similarity rankings respectively; sites that are naturally favored by computer science students as they contain information that is of practical use to them.

**Table 2.** Average classification accuracy between HiBO and Bayesian classifiers

Topics	HiBO classifier	Bayesian classifier	SVM classifier
Dance	97.05%	69.46%	71.58%
Music	94.37%	74.38%	78.49%
Artists	86.45%	83.59%	82.64%
Photography	81.68%	55.28%	69.03%
Architecture	79.77%	69.89%	72.11%
Art History	93.33%	78.47%	68.58%
Comics	95.45%	29.46%	45.24%
Costumes	89.06%	72.43%	69.77%
Design	90.79%	69.29%	55.08%
Literature	89.70%	59.26%	49.91%
Movies	94.59%	71.04%	68.97%
Performing Arts	87.34%	68.08%	65.06%
Collecting	92.87%	67.17%	53.88%
Writing	91.84%	69.56%	60.42%
Graphics	92.68%	79.80%	71.53%
Drawing	91.34%	59.55%	58.16%
Plastic Arts	90.86%	64.36%	62.07%
Mythology	93.58%	68.22%	64.93%
	<b>90.70%</b>	<b>67.18%</b>	<b>64.85%</b>

**Table 3.** Ordering bookmarks for spam

HiBO DR	HiBO Similarity	Google
Block Referrer Spam	Witchvox Article – That Pesky and Obnoxious Spam	www.spam.com
Referrer Log Spamming	Outlook Express Tutorial: Filter- how to stop spam	Fight Spam on the Internet
Spam Assassin	Message Cleaner – Stop viruses and spam emails now	Spam-Wikipedia
Stop Spam with Sneakmail 2.0	The Spammmeister guide to spam	E-mail Spam-Wikipedia
Anti-Spam	Spamhuntress – Spam Cleaning for Blogs	FTC-Spam-Home Page
A Plan for Spam	Discuss Sam Forums-Learn how to eliminate and prevent spam	Coalition Against Unsolicited Commercial Email
Death to Spam	SpamFixer	SpamAssassin
Spam Filter for Outlook	Spam Email Discussion List	Spam Cop
The Spam Weblog	Emailabuse.org	What is Spam- Webopedia
Damn Spam	Spamcop.net	Spam Laws

## 5 Related Work

Bookmarks are essentially pointers to URLs that one would like to store in a personal Web repository for future reference and/or fast access. Today there exist many commercial bookmark management tools<sup>2</sup>, providing users with a variety of functionalities in an attempt to assist them organize the list of their Web favorites [2] [3] [4] [5]. With the recent advent of social bookmarking, bookmarks<sup>3</sup> “*have become a means for users sharing similar interests to locate new websites that they might not have otherwise heard of; or to store their bookmarks in such a way that they are not tied to one specific computer*”. In this light, there currently exist several Web sites that collect, share and process bookmarks. These include *Simpy*, *Furl*, *Del.icio.us*, *Spurl*, *Backflip*, *CiteULike* and *Connotea* and are reviewed by Hammond et al. [9]. Such social networks of bookmarks are being perceived as recommendation systems in the sense that they process shared files and, based on a combinational analysis of the files themselves and their contributors in the network, they suggest to other network members interesting sites submitted by a different community member. From a research point of view, there have been several studies on how shared bookmarks can be efficiently organized to serve communities. The work of [21] falls in this area and introduces *GiveALink*, an application that explores semantic similarity as a means to process collected data and determine similarity relations among all its users. Likewise, [10] suggest a novel distributed collaborative bookmark system that they call *CoWing* and which aims at helping people organize their shared bookmark files. To that end, the authors introduce the utilization of a bookmark agent, which *learns* the user strategy in classifying bookmarks and based on that knowledge it fetches new bookmarks that match the local user information need. In light of the above, we perceive our work on HiBO to be complementary to existing approaches. However, one aspect that differentiates our system from available bookmark management systems is that HiBO provides a built-in subject hierarchy that enables the automatic classification of bookmark URLs on the side of either an individual user or group of users. Through the subject hierarchy, HiBO ensures the dynamic maintenance of personalized views to shared files and as such it assists Web users share their information space with the community.

## 6 Concluding Remarks

In this paper we presented HiBO, a bookmark management system that automatically manages orders, retrieves and mines the data that is either stored in Web users' personal Web repositories or shared across community members. An obvious advantage of our system when compared to existing bookmark management tools is that HiBO uses a built-in subject hierarchy for dynamically grouping bookmarks thematically without any user effort. Another advantage of HiBO is the ordering of bookmarks into the hierarchy's topics in terms of their content importance to the underlying topics. Currently, we are working on privacy issues so as to motivate Web users donate their Web favorites to HiBO and therefore launch a powerful bookmark mining system to the community.

---

<sup>2</sup> For a complete list of available bookmark management systems we refer the reader to [http://dmoz.org/Computers/Internet/On\\_the\\_Web/Web\\_Applications/Bookmark\\_Managers/](http://dmoz.org/Computers/Internet/On_the_Web/Web_Applications/Bookmark_Managers/)

<sup>3</sup> [http://en.wikipedia.org/wiki/Bookmark\\_%28computers%29](http://en.wikipedia.org/wiki/Bookmark_%28computers%29)

## References

1. Abrams, D., Baecker, R. and Chignell, M. Information Archiving with Bookmarks: Personal Web Space Construction and Organization. In Proceedings of the Human Computer Interaction Conference, 1998, pp. 41-48.
2. BlinkPro: Powerful Bookmark Manager <http://www.bookmarksplus.com/>
3. Bookmark Tracker <http://www.bookmarktracker.com/>
4. Check and Get <http://activeurls.com/en/>
5. iKeepBookmarks <http://www.ikeepbookmarks.com/>
6. Open Directory Project: <http://dmoz.org>
7. WordNet 2.0: <http://www.cogsci.princeton.edu/~wn/>.
8. Barzilay, R. and Elhadad, M. Lexical chains for text summarization. In Advances in Automatic Text Summarization. MIT Press, 1999.
9. Hammond, T., Hannay, T., Lund, B. and Scott, J. Social Bookmarking Tools (I): A General Review. D-Lib Magazine, 11(4): doi:10.1045/april2005—hammond, 2005.
10. Kanawati, R., Malek, M., Klusch, M. and Zambonelli F. CoWing: A Collaborative Bookmark Management. In Lecture Notes in Computer Science, ISSN 0302-9743, 2001.
11. Karousos, N., Panaretou, I., Pandis, I. and Tzagarakis, M. Babylon Bookmarks: A Taxonomic Approach to the Management of WWW Bookmarks. In Proceedings of the Metainformatics Symposium 2002, 42-48.
12. Kozanidis, L., Tzekou, P., Zotos, N., Stamou, S., and Christodoulakis, D. Ontology-Based Adaptive Query Refinement. To appear in Proceedings of the 3<sup>rd</sup> International Conference on Web Information Systems and Technologies, 2007.
13. Krikos, V., Stamou, S., Ntoulas, A., Kokosis, P. and Christodoulakis, D. DirectoryRank: Ordering Pages in Web Directories. In Proceedings of the 7<sup>th</sup> ACM International Workshop on Web Information and Data Management (WIDM), Bremen, Germany, 2005.
14. Li, W.S., Vu, Q., Chang, E., Agrawal, D., Hirata, K., Mukherjee, S., Wu, Y.L., Bufi, C., Chang, C.K., Hara, Y., Ito, R., Kimura, Y., Shimazu, K. and Saito, Y. PowerBookmarks: A System for Personalizable Web Information Organization, Sharing and Management. In Proceedings of the ACM SIGMOD Conference, 1999, pp. 565-567.
15. Maarek, Y., and Shaul, I. Automatically Organizing Bookmarks per Contents. In Proceedings of the 5<sup>th</sup> Intl. World Wide Web Conference, 1996.
16. McKenzie, B. and Cockburn, A. An Empirical Analysis of Web Page Revisitation. In Proceedings of the 34<sup>th</sup> Hawaii Intl. Conference on System Sciences, 2001.
17. Page, L., Brin, S., Motwani, R. and Winograd, T. The PageRank citation ranking: Bringing order to the web. Available at: <http://dbpubs.stanford.edu:8090/pub/1999-66>, 1998.
18. Resnik, Ph. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In Proceedings of the 14<sup>th</sup> Intl. Joint Conference on Artificial Intelligence, 2005, pp. 448-453.
19. Stamou, S. and Christodoulakis, D. Integrating Domain Knowledge into a Generic Ontology. In Proceedings of the 2<sup>nd</sup> Meaning Workshop. Italy, 2005.
20. Stamou, S., Ntoulas, A., Krikos, V., Kokosis, P., and Christodoulakis, D. Classifying Web Data in Directory Structures. In Proceedings of the 8<sup>th</sup> Asia-Pacific Web Conference (APWeb), Harbin, China, 2006, pp. 238-249.
21. Stoilova, L., Holloway, T., Markines, B., Maguitman, A. and Mencezer, F. GiveALink: Mining a Semantic Network of Bookmarks for Web Search and Recommendation. In Proceedings of the LinkKDD Conference, Chicago, IL, USA, 2005.