Supporting Collaborative Hierarchical Classification: Bookmarks as an Example

Dominik Benz[1], Karen H. L. Tso[2], Lars Schmidt-Thieme[2]

[1] Computer-based New Media Group (CGNM),
Department of Computer Science, University of Freiburg
Georges-Köhler-Allee 51, 79110 Freiburg, Germany
dbenz@informatik.uni-freiburg.de

[2] Information Systems and Machine Learning Lab (ISMLL),
University of Hildesheim
Samelsonplatz 1, 31141 Hildesheim, Germany
{tso,schmidt-thieme}@ismll.uni-hildesheim.de

May 7, 2007

Abstract

Bookmarks (or Favorites, Hotlists) are a popular strategy to relocate interesting websites on the WWW by creating a personalized URL repository. Most current browsers offer a facility to locally store and manage bookmarks in a hierarchy of folders; though, with growing size, users reportedly have trouble to create and maintain a stable organization structure. This paper presents a novel collaborative approach to ease bookmark management, especially the "classification" of new bookmarks into a folder. We propose a methodology to realize the collaborative classification idea of considering how similar users have classified a bookmark. A combination of nearest-neighbour-classifiers is used to derive a recommendation from similar users on where to store a new bookmark. A prototype system called *CariBo* has been implemented as a plugin of the central bookmark server software *SiteBar*. All findings have been evaluated on a large scale real user dataset with promising results, and possible implications for shared and social bookmarking systems are discussed.

1 Introduction

The continuing, explosive growth of the WWW strengthens its role as a prevalent source of information for scientific research as well as everyday work and leisure. Studies on web usage like [6] reported that revisits are a major part (58%) of website visits. Bookmarks (or Favorites, Hotlists) are a widely used strategy to relocate sites of interest that allows the user to create a personalized URL repository, which facilitates an easy and fast access to relevant information [1]. Traditionally, these repositories are stored on the client-side and can be organized in a hierarchical folder structure via the browser interface. Recently, server-side mechanisms like the so-called "*Social Bookmarking*" have gained popularity [8]. The latter is often characterized by a non-hierarchical classification paradigm known as *Collaborative Tagging*. For the hierarchical case, however, difficulties remain in organizing and maintaining the hierarchical structure with growing size of the repository. An example is the classification of new bookmarks, i.e. finding or creating an appropriate folder to store them.

This paper presents a novel approach to automate the bookmark classification process, aiming at recommending appropriate folders to a user when filing new bookmarks. There are two basic strategies to solve the problem of how to generate such recommendations: the first one, commonly referred to as *information filtering* or *content-based filtering* [14], draws inferences from the user's past behaviour. In this context, "behaviour" means which bookmarks the user stored in which folders.

The second strategy, usually referred to as *collaborative filtering* [14], takes the behaviour of others into account, especially of those who displayed similar interests in the past. In other words, the basic idea is to find similar users who have already classified a bookmark, and then to derive recommendations on where the target user could store this bookmark.

The central contribution of this paper is to present a collaborative classification algorithm for bookmarks. The novelty hereby consists of recommending structural information from similar users. This has scarcely been researched in the context of bookmark classification, where content-based approaches prevail. A prototype of a collaborative bookmark classification system, *CariBo*, has been built, and experimentation results with this prototype and real user data confirm that the presented approach can outperform content-based approaches.

This paper is structured as follows: Section 2 details on general aspects of bookmarking and problems that exist hereby. Section 3 gives an overview of existing work. Our collaborative approach is presented in detail in section 4,

while section 5 presents a content-based approach. Implementation details can be found in section 6. The evaluation procedure and results are given in section 7, and in section 8 we conclude.

2 Bookmarks in general

Studies on web and bookmark usage as well as the current rapid growth of social bookmarking tools like *del.icio.us*[1] (see section 3) suggest that bookmarks are a popular method among users to facilitate the access to WWW information. Abrams [1] as an example cited a survey conducted in 1996 with 6619 web users, where 80% of the subjects reported bookmarks as a strategy for locating information. 92% of them had a bookmark archive, 37% had more than 50 bookmarks. Three years after its foundation in September 2003, *del.icio.us* claimed to serve 1 million users[2].

The studies of Abrams and others [6] confirmed that users tended to have problems with bookmark management, especially when the size of the collection increased. Kanawati and Malek categorized the problems into three classes [10]:

- **Resource discovery**
  The problem of "finding good bookmarks" that match the user's information needs. This is not a purely bookmark-specific problem, but corresponds heavily to the problem of "finding interesting websites". This is addressed by a large research community in the area of recommender systems [14].

- **Recall**
  The problem of locating an appropriate bookmark at a given time.

- **Maintenance**
  The problem of keeping the set of bookmarks up-to-date and well-organized; difficulties hereby arise from discovering broken links, modifying the organization scheme due to changes in personal interest, and creating and maintaining the taxonomy implied by the bookmark folder hierarchy. The classification of new bookmarks also belongs to this category.

Abrams et al. pointed out that the crucial trade-off in bookmarking is between organization costs and future benefit: *"Users must weigh the cost*

[1] http://www.del.icio.us
[2] http://blog.del.icio.us/blog/2006/09/million.html

of organizing bookmarks against the expected gains"[1]. Roughly half of their subjects turned out to be "sporadic filers", i.e. users who occasionally schedule reorganization sessions when their bookmark repository became too complex. This task was generally reported to be time-consuming and tedious. Among others, their implications for the design of a (possibly shared) bookmark management system included to *"provide users with an immediate filing mechanism when creating a bookmark"*. We argue that using a collaborative classification algorithm for this purpose is a sensible choice.

3 Related Work

As bookmarking is one of the most commonly used features on the web, there is a vast number of programs and tools with the purpose to alleviate different aspects of bookmark management. A large number of these tools can be assigned to the category "centrally store and browse", whereby the core benefit is to make bookmarks available when the user moves to another physical machine. This concept is extended in some cases by making bookmarks shareable with other users. Already mentioned above, **del.icio.us**[3] is a popular online service which transfers the usual client side bookmarking mechanism onto a central server to enable roaming and bookmark sharing. This has become known as *social bookmarking.* Instead of a hierarchical classification scheme, each user can tag his bookmarks with a set of arbitrary keywords, facilitating a "by-keyword"-access to own or other users' bookmarks. **Spurl**[4] is an example of a social bookmark service that retains the hierarchical folder structure known from client-side mechanisms. It is important to notice that in both cases, no personalized recommendation takes place on how a particular user could classify a new bookmark. The individual repositories are simply made "browsable".

An example for a more personalized solution to the resource discovery problem is **GroupMark** [13], a WWW recommender system. It takes the users' bookmarks as the primary source of information to assign them to peer recommender groups. From those, they will receive suggestions for potentially interesting websites.

In addition to website recommendation, **InLinx** by Bighini et al. [3] facilitates the automatic classification of bookmarked websites into globally predefined categories. The basis for the classification is the user's profile

[3]http://www.del.icio.us
[4]http://www.spurl.net

and the content of the web page. Two further approaches that use this basis are [12] (employing a semi-automatic clustering algorithm for reorganization of the bookmark hierarchy) and [11] (comparing different document classification methods).

All of the described approaches address different aspects, but leave out an important source of information, namely to consider the bookmark organization habits and strategies of similar users. The focus of this paper is to propose an algorithm to automatically classify bookmarks based on the classifications of similar users. This collaborative methodology of recommending structural information has scarcely been researched in the context of hierarchical classification schemes. Haase et al. [7] presented a more general approach of how the evolution and management of personal ontologies can be supported by a collaborative recommendation algorithm.

4 Collaborative Approach

Pemberton et al. pointed out that the basic idea of collaborative filtering is *"to recruit others to act as our filtering agents on the assumption that they are our peers, i.e. like us in tastes and judgements of quality"* [13]. For the case of bookmark management, one could replace "filtering agents" with "classification agents" or "annotation agents". Different groups of people obviously have different needs and strategies to organize and annotate bookmarks belonging to a certain category. A computer scientist for example might store a bookmark about web development with PHP in a relatively sophisticated hierarchy like *development* > *web development* > *languages* > *PHP*. A sales consultant, however, would probably file the same bookmark in a less differentiated organization scheme, possibly something like *marketing* > *websites*. Analogously, the annotations that these both persons would use for this website will in all probability differ. The computer scientist might annotate the PHP page with something like "*dynamic, script language, LAMPP*", whereas one could imagine annotations like "*advanced webdesign, programming, webserver*" for the sales consultant.

Consequently, having a look in our peer group, i.e. people who are interested and engaged in similar topics as we are, is highly probable to give us valuable information how to classify and annotate our own bookmarks. The system described in this paper aims at generating two substantially separate recommendations: Keyword recommendations on the one hand, i.e. which keywords to use for annotating a new bookmark, and a recommendation of a classification on the other hand. We will focus on the latter aspect in the

following discussion; refer to [2] for details on the keyword recommendation.

4.1 Data Model

In order to facilitate the measurement of different types of similarity, we have chosen a uniform representation of the following three basic entities in the system:

- **Links**, i.e. the actual "bookmarks", consisting of a URL, and optionally a title and a description
- **Folders** that contain the bookmarks, labelled with a folder title and optionally annotated with a folder description
- **Users** that own a hierarchy of folders, optionally annotated with a user description

WWW recommender systems like [3] often examine the complete content of websites as data foundation and analyze it with information retrieval techniques. Instead, the presented approach relies on information extracted from the bookmarked URL itself and manually assigned annotations (title, description). One of the main reasons for this decision was that analyzing possibly large HTML documents might slow down the classification process significantly. This would have detrimental effects especially for an everyday task like bookmarking. However, if an analysis technique is able to extract highly descriptive keywords from the actual page content, it can be expected to further improve the recommendation quality of the presented algorithm. For the scope of this work, this has to be left for future work.

4.1.1 Term Vector Space

For data representation, the vector space model, a popular information filtering model for textual material, is used [15]. It has been widely tested and is expressive enough to describe the information content available. Furthermore, it allows in combination with an appropriate database design for a fast computation of recommendations or profile updates, which is crucial to an everyday task like bookmarking.

In the vector space model, links, folders and users are described by a profile vector. Each term that occurs in any title or description in the system adds one dimension to the vector space. In addition, each hostname occurring in any URL adds one more dimension. The normalized term

frequency was used as weight for each term. The dimensionality of the vector space was reduced by stemming, a procedure that tries to reduce the keywords to their word stems. We used *Porter Stemming*, a popular stemming algorithm for the English language [3], which removes suffixes based on a set of condition/action rules that specify, for example, how to remove the plural-"s" from plural terms. Additionally, very common words or terms with little information content ("and, or", etc ...) were removed by using a stopword list[5] containing 429 common English words. We modified this list by adding stopwords belonging to the area of the WWW, e.g. *index, home, homepage, website*. After this, the list contained 460 entries.

4.1.2 Taxonomy Representation

To represent the hierarchical organization of a bookmark collection, the terms were aggregated in a bottom-up manner through the taxonomy tree. Starting from the links as "leaves", all folders inherit all terms and the corresponding frequencies of their contained links. Then, all parent folders inherit all terms and frequencies describing their descendant folders, up to the user's root folder. The user profile itself inherits all terms and frequencies of the user's root folder. Hence, the profile of a folder becomes more general the closer it is to the hierarchy root. We argue that this simple mechanism reflects the intuitive organization principle of increasing folder specificity with increasing depth in the hierarchy. This is why we consider this aggregation as a sufficient representation of the hierarchical structure. Furthermore, the additional storage and computational consideration of the graph structure itself might lead to a complexity overhead hardly justified in relation to the possible benefits.

4.1.3 Similarity Measure

To measure similarity between two profile vectors, this approach uses the cosine vector similarity, a common measure in the context of the vector space model. It defines the similarity of two profile vectors, $prof_x$ and $prof_y$, as the cosine of the angle between them and can be computed as:

$$sim(prof_x, prof_y) = \frac{prof_x \cdot prof_y}{|prof_x||prof_y|}$$

Obviously, for the computation of the dot product in the numerator of this fraction, only those entries that have a value greater than zero in

[5] http://www.lextek.com/manuals/onix/stopwords1.html

both vectors are relevant. In combination with computing and storing the norm at vector creation time, this allows for an efficient computation of this similarity measure, considering only the intersection of the keyword sets of two entities (links, folders or users).

The uniform vector representation of links, folders and users in combination with the mentioned similarity measure provides us the ability to measure various relations inside our domain. First of all, we can measure the similarity between two users, two folders or two links. But similarities can also be computed between different entities, e.g. between a link and a folder.

4.2 Classification Process

Given that a similar user has already bookmarked a certain URL in one folder of his bookmark hierarchy, the basic problem consists in mapping the location of this folder to a folder location in the target user's bookmark hierarchy. This can be seen as a problem of taxonomy mapping. Another aspect that needs to be considered is what to do if we do not find such a corresponding folder. As there is no approach of collaborative classification found in the research area of bookmark management, it is difficult to draw comparisons or to point out the predominance of the approach presented here. This is why we have implemented a content-based classification algorithm as well as a random algorithm to compare the results (see section 5).

Figure 1 gives an overview of the process of collaborative classification. The figure is to be read from left to right. It depicts the process when a user u adds a new bookmark l. The first step is to find similar users in the system that have already bookmarked l. $U_{sim,l}$ is the set of those users, sorted in descending order by user similarity. Two parameters control the size of the group: (i) The maximal number of similar user to consider; (ii) the similarity threshold to which extent a user is considered to be similar. Table 1 contains the values used for the evaluation.

$F_{sim,l}$ contains all folders in which the users from $U_{sim,l}$ have stored the link l. Assuming that there are no URL duplicates for each user, it is obvious that $|U_{sim,l}| = |F_{sim,l}|$.

For each of the folders in $F_{sim,l}$, we now try to find the most similar folders of user u himself. This results again in a set of folders $F_{rec,l}$, containing only folders owned by user u. Two parameters control the cardinality of $F_{rec,l}$: (i) The number of similar folders to be considered for each folder $f_{sim} \in F_{sim,l}$; (ii) the folder similarity threshold to which extent a folder is

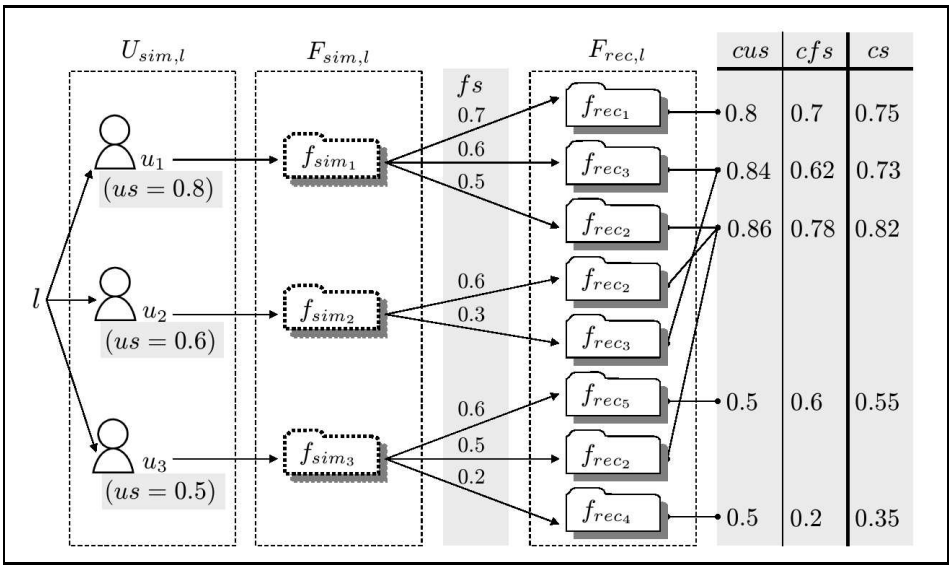


Figure 1: Overview of the collaborative classification process. Abbreviations used: $us = sim(u, u_i)$, $fs = sim(f_{sim_i}, f_{rec_j})$ $(i \in \{1, 2, 3\}, j \in \{1, \ldots, 5\})$; $cs = \frac{cus+cfs}{2}$

Parameters controlling the size of $U_{sim,l}$:

Nr. of similar users to consider	3
Threshold of user similarity	0.1

Parameters controlling the size of $F_{rec,l}$:

Nr. of similar folders to consider	3
Threshold of folder similarity	0.01

Parameter controlling new folder creation:

Threshold when to create a new folder	0.2

Table 1: Parameters controlling the collaborative classification and values used for the evaluation

considered to be similar (see table 1).

For the purpose of finding the best folder recommendation among $F_{rec,l}$, we can consider three variables (as explained above): (i) The similarity of the recommending user (denoted as *us* in the diagram); (ii) the folder similarity of his folder with our corresponding folder (denoted as *fs*); (iii) the number of times a folder has been recommended, i.e. how often it occurs in $F_{rec,l}$.

The following ideas of how to combine them are intuitive:

- *Choose the folder which has been recommended most often.* This completely neglects user and folder similarities, and is hence insufficient. If folder A is recommended by 3 marginally similar users and folder B by 2 very similar users, folder A would be the choice - which is not the desired behaviour.

- *Sum up the user and folder similarities for each folder.* Once again, this would lead to a strong domination of folders that were recommended often, with the same disadvantages just mentioned.

- *Average the user and folder similarities for each folder.* Hereby, the number of times a folder has been recommended would lose influence. If folder A has been recommended 10 times, but its average similarity values are slightly smaller than the ones of folder B who has been recommended just once, this approach would wrongly choose to recommend B. If there is one very similar user who happens to have a very similar folder, this user would strongly dominate the recommendation process.

We argue that a combination of the above ideas is required that strikes the balance between the number of times a folder has been recommended and the average similarity values. In this way, the effect of dominating users or folders like in the given examples would be smoothed. Of course, this becomes necessary only when a folder has been recommended by more than one user. In the area of collaborative filtering, usually ratings for certain items are predicted, e.g. by computing a weighted sum of other users' votes [4]. In our case, we want to predict classifications instead of ratings; so we have adapted this technique to compute for each recommended folder a *combined user similarity* of all users who have recommended it (denoted *cus* in the diagram) and a *combined folder similarity* (*cfs*) of all folders the recommended folder has been mapped from. Taking the combined user similarity for the recommended folder f_{rec} as an example, the following values are used for its computation:

- the average user similarity of all users who recommended f_{rec} (denoted $avg_{U_{sim,l,f_{rec}}}$ below)
- the number of users that recommended f_{rec} (denoted $|U_{sim,l,f_{rec}}|$ below)
- the total number of recommending users (denoted $|U_{sim,l}|$ below)

The core idea is to weigh the average by the proportion of all recommending users who have recommended the particular folder f_{rec}. Notated formally, the combined user similarity, $cus_{f_{rec}}$, the combined folder similarity, $cfs_{f_{rec}}$, and the final combined similarity value $cs_{f_{rec}}$ of a recommended folder, f_{rec}, to user u are computed according to:

$$cus_{f_{rec}} = avg_{U_{sim,l,f_{rec}}}(1 + (1 - avg_{U_{sim,l,f_{rec}}})\frac{|U_{sim,l,f_{rec}}|}{|U_{sim,l}|})$$

$$cfs_{f_{rec}} = avg_{F_{sim,l,f_{rec}}}(1 + (1 - avg_{F_{sim,l,f_{rec}}})\frac{|F_{sim,l,f_{rec}}|}{|F_{sim,l}|})$$

$$cs_{f_{rec}} = \frac{cus_{f_{rec}} + cfs_{f_{rec}}}{2}$$

Whereas

- Average user and folder similarity:

$$avg_{U_{sim,l,f_{rec}}} = \frac{1}{|U_{sim,l,f_{rec}}|} \sum_{u_{sim} \in U_{sim,l,f_{rec}}} sim(u_{sim}, u)$$

$$avg_{F_{sim,l,f_{rec}}} = \frac{1}{|F_{sim,l,f_{rec}}|} \sum_{f_{sim} \in F_{sim,l,f_{rec}}} sim(f_{sim}, f_{rec})$$

- Set of all similar users that would recommend to put link l in folder f_{rec}:

$$U_{sim,l,f_{rec}}$$

- Set of all folders containing l mapped to the recommended folder f_{rec}:

$$F_{sim,l,f_{rec}}$$

The resulting combined similarities are found in the rightmost columns of figure 1. The final combined similarity of a recommended folder (denoted *cs* in the diagram) is computed as the mean of its combined user similarity

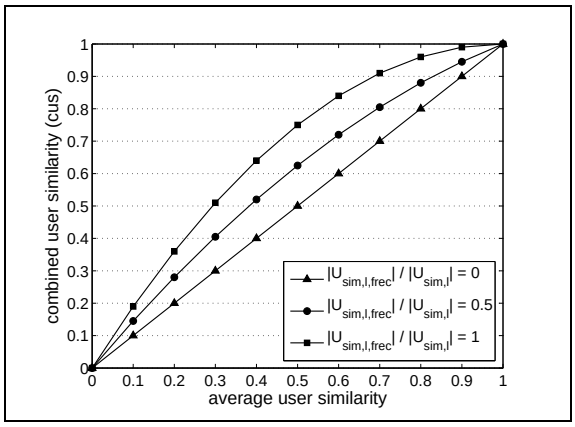


Figure 2: Weighting effect on the combined user similarity

and its combined folder similarity. In the example, folder f_{rec_2} would be recommended with a final similarity value of 0.82.

Figure 2 visualizes the weighting effect on the average user similarity (found on the x-axis) of a folder f_{rec}. The higher the proportion $\frac{|U_{sim,l,f_{rec}}|}{|U_{sim,l}|}$ of users who have recommended f_{rec} for link l, the higher the combined similarity value. Please note that $\frac{|U_{sim,l,f_{rec}}|}{|U_{sim,l}|} = 0$ can never be reached and is just included as a lower boundary.

To establish a connection to standard classification methods, this approach can be considered as an application of a two-step k-nearest-neighbour-classifier (k-NN). Usually, this algorithm is used for document classification. A new document is classified to the category the majority of its k most similar documents in the training set belong to. The approach described above considers in the first step a set of k similar users who have bookmarked a link l (forming the set $U_{sim,l}$). In the second step it considers for each of those users a set of k similar folders of the target user (forming the set $F_{rec,l}$). So the first step can be regarded as classifying the current user into a group of interest, the second step as classification of the current bookmark according to the needs and habits of that group. For the evaluation, a value for k of 3 has been chosen.

4.2.1 Creating new folders

The methodology described above will perform best if the current user already has an existing folder with a sufficient total similarity to a new bookmark. If the latter is not the case, a recommendation is desirable to create a

new folder, mainly concerned with (i) how to label the new folder, (ii) where to place it in the target user's hierarchy and (iii) to which degree ancestors of the new folder should also be created.

Generating such a recommendation is a very interesting task, but is not the main focus of this paper at this time. Hence we have excluded this aspect in the evaluation. Nevertheless we have implemented a methodology, that might serve as basis for future work and evaluation. We propose to recommend to create a new folder under the following conditions: (i) In the case when the recommended folder happens to be the target user's root folder. Storing a link in this root folder would not contribute to increase the level of organization. (ii) In the case when the total similarity of the recommended folder falls below a certain threshold. Then it can be assumed that the content of this folder is somehow related to the new bookmark, but is probably not a very specific match. Creating a new subfolder inside this folder can be considered as an appropriate way to create a more specific storage location.

As a folder recommendation might stem from several users having folders with different names, a decision is necessary which name to recommend as a label for the folder to be created. For this approach we adapted the most intuitive idea to use the name the most similar of the recommending users has used. The final question is to which extent a hierarchy of folders is to be created. Before recommending to create a new folder, its ancestors are checked for similarity with the target folder. If a higher value is found, it is recommended to create a hierarchy with appropriate depth.

Besides the proposed collaborative approach, the next section presents a content-based approach for bookmark classification.

5 Content-based Approach

As detailed at the beginning of this section, another source of information to reason about automatic bookmark classification is the target user's own classification history. This is a classical case of content-based recommendation. We have implemented this algorithm for comparison reasons. In a prior study, we also considered a public directory as another source of information, yielding rather poor recommendation results [2]. For further comparison, we also implemented a random classification algorithm.

5.1 User's classification history

Having the vector space model described above at hand, finding the best existing folder for a new bookmark can be done in a straightforward manner. First, a profile vector for the new link is generated, based on the URL as well as title and description the user has assigned to it (eventually supported by meta information found in the page content). The resulting profile vector is compared with the profile vectors of all existing folders. The most similar folder is recommended. This is a typical application of a nearest-neighbour-classifier (NN). A requirement for a reliable recommendation is the existence of a sufficiently similar folder. The abilities of this method to recommend the creation of new folders are very limited.

5.2 Random Algorithm

For comparison reasons, we implemented an algorithm that randomly recommends a folder of the target user for a new bookmark.

6 Implementation

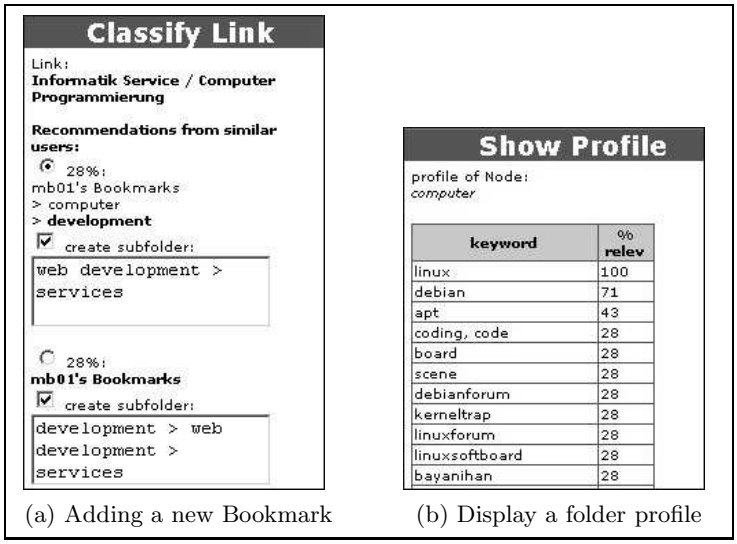
The prototype implementation is called *CariBo* (**C**oll**a**bo**r**at**i**ve **Bo**okmark Classifier) and is based on the open source bookmark server *SiteBar*[6]. SiteBar as a sourceforge-project is an open-source software written in PHP to centrally store and share bookmarks on a webserver. All system data is stored in a MySQL database. The implementation was done using PHP 5.0.4 along with MySQL 4.0.21 and was tested on a machine equipped with a 2.8 GHz Intel Xeon Processor, 2 GB RAM and the SuSE Linux Operating System (version 9.3). Figure 3a shows the user interface where the outcome of the collaborative classification is presented to the user, Figure 3b depicts the display of a folder profile. Installation instructions and downloads can be found at our group website[7].

7 Experimental Evaluation

In a prior case study with 15 subjects, our collaborative approach clearly outperformed the content-based one [2]. Encouraged by these results, we used a

[6] http://www.sitebar.org

[7] http://www.informatik.uni-freiburg.de/cgnm/software/caribo/index_en.html



(a) Adding a new Bookmark (b) Display a folder profile

Figure 3: Screenshots of the user interface

General

Nr. of test users	619
Total Nr. of bookmarks	206365
Total Nr. of distinct URLs among bookmarks	155736
Min / Max / Average nr. of bookmarks per user	5 / 7364 / 332.18
Min / Max / Average nr. of folders per user	1 / 1024 / 41.6
Average nr. of bookmarks per folder	7.99

Extracted terms and profile vectors

Total nr. of terms in database after initializing all profiles	135208
Nr. of English terms[a]	15709 (11.6%)
Nr. of numeric terms	3020 (2.2%)
Nr. of domain names among terms	67040 (49,6%)
Nr. of other terms	49439 (36.6%)
Min / Max / Average nr. of terms assigned to a bookmark	1 / 34 / 4.15
Min / Max / Average nr. of terms assigned to a folder (after bottom-up keyword inheritance, see section 4.1.2)	1 / 10735 / 60.82

[a]counted by looking up the terms in a MySQL version of the WordNet 2.0 database (http://www.androidtech.com/html/wordnet-mysql-20.php)

Table 2: Experimentation data statistics

large-scale real-user dataset crawled by Herold [9] from the social bookmarking platform *spurl.net*[8] for further evaluation. As mentioned in section 2, Spurl enables each user to mark his bookmarks as "private" or "public" and to store them in a hierarchical folder structure. Herold crawled the public bookmarks of all users visible in the "discover users" - section along with the categories they were organized in. From this initial dataset, we included each user who had bookmarked at least one of the 200 most bookmarked URLs, leading to a set of 619 test users. As Spurl allows several root folders, we added an artificial single root folder for each user and appended his original root folders as subfolders. After the users along with their complete bookmark collections were imported into our system, the profile for each test user was initialized. Refer to table 2 for further statistics. The parameters controlling the collaborative classification were set according to table 1. The given values were taken over from the prior case study. It is subject to further investigation how to optimize the classification performance by modifying those parameters.

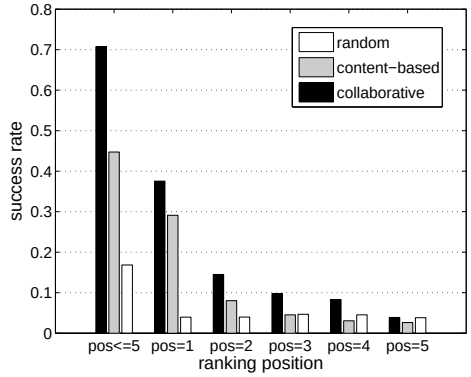
For each test user and each of his test links (i.e. the intersection of his URL collection and the 200 most bookmarked URLs), a "leave-one-out"-testing was applied: the current test link was removed from his collection, and given to the three classification algorithms (collaborative, content-based, random). This led to a total of 5015 classification decisions for each algorithm. The outcome of each algorithm was a list with the top 5 recommendations of folders where the user could classify the bookmark. A classification was judged as a hit when the algorithm recommended the folder where the bookmark was taken out from. For each ranking position, the success rate was measured as the proportion of hits among all classifications. Figure 4 displays the results[9].

7.1 Discussion

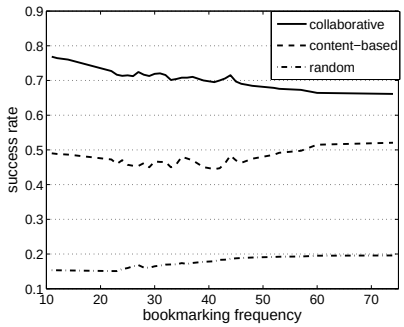
With a success rate of 70.73% among the top five recommended folders (denoted as $pos \leq 5$ in figure 4a), the collaborative algorithm clearly outperforms the content-based approach (44.73%). This predominance is retained on every ranking position (denoted as $pos = 1, pos = 2, \ldots$). The random algorithm shows a nearly constant hit rate of around 4% across all positions. The average time needed to generate a collaborative recommendation (1.82 seconds) is significantly higher than for the content-based algorithm

[8] http://www.spurl.net

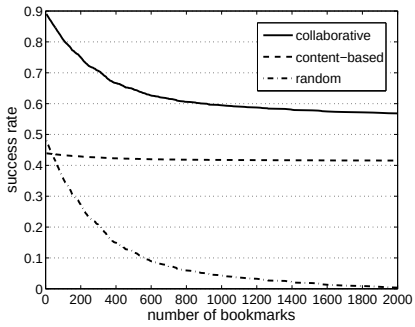
[9] To enhance clarity, the values for figure 4b and figure 4c were smoothed with a Savitzky-Golay filter using polynomial order 1 and frame size 21 and 361, respectively.



(a) Success Rate on ranking position *pos*



(b) Success Rate on top five ranking positions ($pos \leq 5$) depending on the number of users that have bookmarked a certain URL (bookmarking frequency)



(c) Success Rate on top five ranking positions ($pos \leq 5$) depending on the size of bookmark repositories (number of bookmarks)

Figure 4: Evaluation Results

(0.38 seconds), but can be seen as justified regarding the observed gain of recommendation quality.

For the collaborative algorithm, an interesting question is how many users need to be present in the system that have already bookmarked a certain URL in order to generate a recommendation. Figure 4b displays the success rates depending on the bookmarking frequency, i.e. the number of times a certain URL has been bookmarked. Although slightly decreasing for the collaborative algorithm, the performance is rather stable throughout all frequencies. This implies that already a small number of other people who have bookmarked a certain URL can serve as a sufficient basis for a successful recommendation. Figure 4c compares the performance depending on the different bookmark repository sizes, i.e. the number of bookmarks inside a particular collection. Hereby the content-based algorithm seems to perform equally throughout all bookmark repository sizes. The random algorithm performs naturally better on smaller repositories, as the probability of a hit increases with decreasing repository size. Interestingly, the success rate of the collaborative algorithm increases in a similar fashion, which means that it is able to produce sensible recommendations also for users with a relatively small set of bookmarks. Of course small bookmark repositories are easier to manage manually than larger ones. But nevertheless these results corroborate the general idea of collaborative classification.

Kim [11] reported success rates between 53% and 59% for hits purely content-based k-nearest-neighbour classifier approach, based on 400 to 800 training documents. In this method, also the text content of the websites was analyzed. Comparing this to the relatively sparse information available for each bookmark in our case (roughly 4 terms, see table 2), the advantages of the collaborative classification idea become clear: the sparsity of the information inside a single personal bookmark repository can be counterbalanced when it is shared with a larger community and information of similar users is taken into account. Interestingly, one can assume that the contributors have structured their repositories only for personal use, e.g. not adhering to a standard vocabulary to maximize the benefit for the community. Nevertheless, the aggregation of genuinely self-centered structuring activity can lead to a benefit for the community, as the results of our evaluation suggest. Dynamics like these have also been observed in the context of collaborative tagging [5]. It is notable at this point that the Spurl test dataset was naturally limited to bookmark folders which were not marked as private by their owners. From our results, we speculate that the inclusion of these folders would not make a big difference; but due to the inherent privacy restrictions, this has to remain a hypothesis.

8 Conclusions and Outlook

This paper presented a novel approach to automatic bookmark classification, based on the classifications of similar users. The main method presented was an application of a k-NN-classifier to generate collaborative recommendations for classifying new bookmarks. The latter uses a weighted average technique to regulate the influence of several users.

The central contribution of this paper is to demonstrate that the classifications of other users, especially similar ones, are a valuable source of information for an automatic bookmark classification process that should not be neglected when designing shared bookmark systems. In the presented evaluation, the collaborative classification outperformed clearly the content-based approach and compared well to other studies. Especially as social bookmarking systems like *del.icio.us* gain popularity, the results of this study open a new perspective on extending the functionality of such shared bookmark repositories.

Nevertheless, another result is that collaborative classification alone cannot be seen as the golden mechanism that relieves a user from all tasks of bookmark organization. The user cold start problem inherent to all recommender systems is alleviated when users submit their bookmarks when joining the system. But the system cold start problem is more critical, as recommendations from other users require other users to be present in the system. For further research it could be promising to examine how synergies can arise from combining the results of invoking additional sources for automatic classification, i.e. the user's classification history and public directories - the first used only for comparison here.

Another promising direction for further improvement of the presented approach is to provide users with mechanisms to control the taxonomy structure. If a user defines e.g. a maximum depth level of the taxonomy, a maximum number of folders or a maximum number of bookmarks per folder, clustering techniques might help to support the process of splitting or merging folders.

Long-term experimentation with direct user feedback on the classifications can be expected to further prove the utility of collaborative classification. Another critical aspect of the real-world application is how to ensure privacy, e.g. by a control mechanism which folder or link information should be available for others. *SiteBar*, the base system of the presented implementation, already offers the basic access control features that could be extended.

References

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

[2] D. Benz, K. H.-L. Tso, and L. Schmidt-Thieme. Automatic bookmark classification - a collaborative approach. In *Proceedings of the 2nd Workshop in Innovations in Web Infrastructure (IWI2) at WWW2006*, Edinburgh, Scotland, May 2006.

[3] C. Bighini, A. Carbonaro, and G. Casadei. Inlinx for document classification, sharing and recommendation. In V. Devedzic, J. M. Spector, D. G. Sampson, and Kinshuk, editors, *Proc. of the 3rd Int'l. Conf. on Advanced Learning Technologies*, pages 91–95. IEEE CS, Los Alamitos, CA, USA, 2003.

[4] J. S. Breese, D. Heckerman, and C. M. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *UAI*, pages 43–52, 1998.

[5] C. Cattuto, V. Loreto, and L. Pietronero. Collaborative tagging and semiotic dynamics, 2006.

[6] A. Cockburn and B. McKenzie. What do web users do? an empirical analysis of web use. *International Journal of Human-Computer Studies*, 54:903–922, 2002.

[7] P. Haase, A. Hotho, L. Schmidt-Thieme, and Y. Sure. Usage-driven evolution of personal ontologies. In *Proceedings of the 3rd International Conference on Universal A ccess in Human-Computer Interaction (UAHCI)*, Las Vegas, Nevada USA, 22-27 July 2005.

[8] T. Hammond, T. Hannay, B. Lund, and J. Scott. Social bookmarking tools (i) - a general review. *D-Lib Magazine*, 11(4), April 2005. ISSN 1082-9873.

[9] M. Herold. Collaborative personal ontology evolution. Diploma thesis, Albert-Ludwigs-University, Freiburg, Germany, December 2005.

[10] R. Kanawati and M. Malek. Informing the design of shared bookmark systems, 2000.

[11] I.-C. Kim. A personal agent for bookmark classification. In Y. S.-T. Y. M, editor, *Intelligent Agents: Specification, Modeling, and Applications. 4th Pacific Rim International Workshop on Multi-Agents, PRIMA 2001. Proceedings (Lecture Notes in Artificial Intelligence Vol.2132)*, pages 210–21, Dept. of Comput. Sci., Kyonggi Univ., Suwon, South Korea, 2001. Springer-Verlag.

[12] Y. S. Mareek and I. Z. B. Shaul. Automatically organizing bookmarks per contents. *Proc. Fifth International World Wide Web Conference*, May 6-10 1996.

[13] D. Pemberton, T. Rodden, and R. Procter. Groupmark: A WWW recommender system combining collaborative and information filtering. In *Proceedings of the 6th ERCIM Workshop on 'User Interfaces for All'*, number 12 in Long Papers, page 13. ERCIM, 2000.

[14] P. Resnick and H. R. Varian. Recommender systems. *Commun. ACM*, 40(3):56–58, 1997.

[15] G. Salton. *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.