Automatic Bookmark Classification: A Collaborative Approach

Dominik Benz, Karen H. L. Tso, Lars Schmidt-Thieme
Computer-based New Media Group (CGNM)
Department of Computer Science, University of Freiburg
Georges-Köhler-Allee 51, 79110 Freiburg, Germany
{dbenz, tso, lst}@informatik.uni-freiburg.de

ABSTRACT

Bookmarks (or Favorites, Hotlists) are a popular strategy to relocate interesting websites on the WWW by creating a personalized local URL repository. Most current browsers offer a facility to store and manage bookmarks in a hierarchy of folders; though, with growing size, users reportedly have trouble to create and maintain a stable taxonomy. 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. Additionally, a procedure to generate keyword recommendations is proposed to ease the annotation of new bookmarks. A prototype system called *CariBo* has been implemented as a plugin of the central bookmark server software *SiteBar*. A case study conducted with real user data supports the validity of the approach.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information Filtering*; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Intelligent agents*

Keywords

WWW, bookmark, classification, collaborative filtering, recommender systems, augmented averaging

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 [5] 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]. Most current browsers support hierarchical bookmarking schemes that enable users to manage their collection of URLs in a hierarchy of folders. However, with growing size of the repository, difficulties in organizing and maintaining the hierarchical structure arise, as for example 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* [13], 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* [13], 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.

2. BOOKMARKS IN GENERAL

Studies on web usage in general and bookmark usage suggest that bookmarks are a very popular method among users to facilitate the access to WWW information; [1] cites 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.

Cockburn and McKenzie reported an average number of 184 bookmarks within their 17 subjects, organized in a mean of 18.1 folders [5]. Both studies 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 [7]:

- **Resource discovery**
  The problem of "finding good bookmarks" that match

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

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.

- **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. point out that the crucial tradeoff in bookmarking is between organization costs and future benefit: *"Users must weigh the cost 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 is generally reported to be time-consuming and tedious. Among others, their implications for the design of a (possibly shared) bookmark management system include 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 of web browsers, there is a vast number of programs and tools with the purpose to alleviate different aspects of bookmark management. The majority of them 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. **Delicious** (*http://www.del.icio.us*) for example is a popular online service which transfers the usual client side bookmarking mechanism onto a central server to enable roaming. This has become known as *social bookmarking* and has gained popularity recently. Furthermore, the bookmarks can be tagged with a set of keywords, facilitating a "by-keyword"-access to own or other users' bookmarks. It is important to notice that hereby neither classification takes place nor reasoning or recommendation, i.e. whether a certain bookmark might be interesting for a particular user. The individual repositories are simply made "browsable".

An example for a much more personalized solution to the resource discovery problem is **GroupMark** [12], 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. [2] facilitates the automatic classification of bookmarked websites into globally predefined categories. The basis for the classification is the user's profile and the content of the web page. Two further approaches that use this basis are [9] (employing a semi-automatic clustering algorithm for reorganization of the bookmark hierarchy) and [8] (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. Haase et al. [6] presented a more general approach how the evolution and management of personal ontologies can be supported by a collaborative recommendation algorithm.

4. COLLABORATIVE APPROACH

Pemberton et al. point 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"* [12]. 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. Invoking a public directory (like Yahoo or the Open Directory Project) for this purpose is an alternative. But those public taxonomies are apparently either too detailed or not detailed enough. For the computer scientist, Yahoo's categories in the area of webdesign might be not fine-grained enough, whereby they are much too detailed for our sales-consultant. Nevertheless, public directories constitute a source of information worth to be consulted in the absence of other alternatives. Table 1 gives an overview of the conditions for different types of recommendations.

For the reasons given above, 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.

4.1 Data Model

In order to enable any kind of reasoning or recommendation, the following three basic entities in the system need a common representation:

- **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

source of information	*conditions for recommendation*
classifications of similar users	at least one similar user must have bookmarked the same website
user's classifications so far	at least one existing folder needs to be annotated with a keyword present in the new bookmark's profile
public directory	the website needs to be present in the public directory

Table 1: conditions for recommendations based on different information sources

- **Users** that own a hierarchy of folders, optionally annotated with a user description

4.1.1 Data Foundation

WWW recommender systems like [2] often examine the complete content of websites 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). In a limited way, meta-information extracted from the website content is used to find default values for the bookmark title and its description. The main reasons for those decisions were:

- Manually assigned annotations, especially those coming from a peer recommender group, are generally more trustworthy than an automatically generated description.
- Filing a bookmark is a time-critical task. Parsing and analyzing possibly large HTML documents bears the risk of delaying this process. This could have a detrimental effect on a user's adherence to the system.

4.1.2 Term Vector Space

For data representation, the vector space model, a popular information filtering model for textual material, is used [14]. 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 [2], which removes affixes based on a set of condition/action rules that specify, for example, how to remove the plurals from plural terms. Additionally, very common words or terms with little information content ("and, or", etc ...) were removed by using a stopword list [10] containing 429 common English words. It has been modified by adding stopwords belonging to the area of the WWW, e.g. *index, home, homepage, website.* After this, the list contained 460 entries.

4.1.3 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.

4.1.4 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 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 hardly possible to draw comparisons or to point out the predominance of the approach presented here. This is why we have implemented two content-based classification algorithms 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,

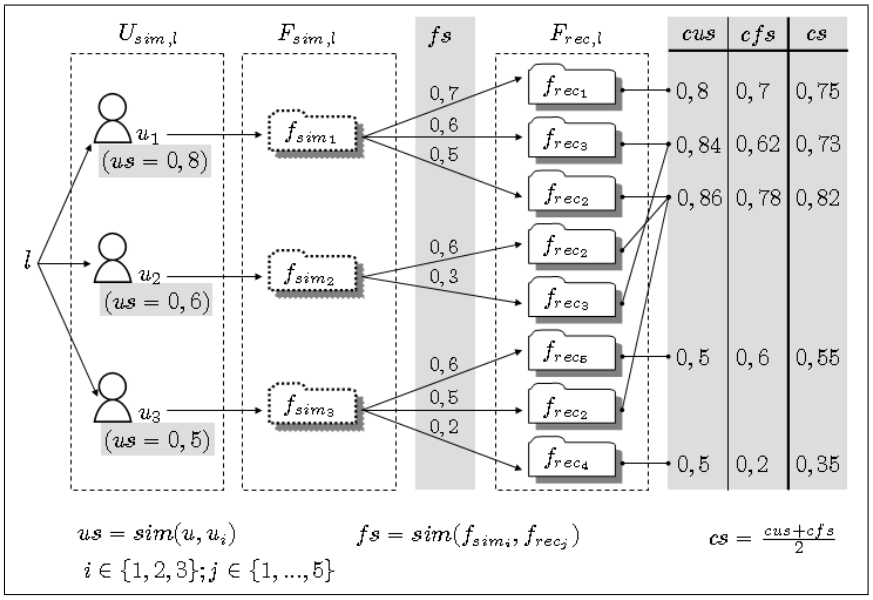


Figure 1: Overview of the collaborative classification process

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	0.3

Table 2: Parameters controlling the collaborative classification and values used

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 2 contains the values used for the case study.

$F_{sim,l}$ 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 considered to be similar (see table 2).

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.

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 individual similarity values. In this way, the effect of dominating users or folders like in the given examples would be smoothened. 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 [3]. Those techniques seemed inappropriate to us for predicting a classification. Therefore, we suggest 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. For both values, we propose to apply a technique named "augmented averaging". Taking the combined user similarity as example, the basic idea behind augmented averaging is to take the average user similarity as starting point, and then to augment it depending on:

- the number of times it has been recommended
- the user similarities
- the total number of recommending users

The maximal possible amount of augmentation is

$$1 - average\_user\_similarity.$$

This maximal amount is divided by the total number of recommending users, and then for each of those fractions, a portion corresponding to a recommending user's similarity is added to the average user similarity.

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

Notated formally, the combined user similarity, $cus_{f_{rec}}$, and the combined folder similarity, $cfs_{f_{rec}}$, of a recommended folder, f_{rec}, to user u are computed according to:

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

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

Whereas

- Average user similarity:

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

- Average folder similarity:

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

- 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}:

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

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 case study, 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 highly desirable to create a new folder, mainly concerned with

- how to label the new folder,
- where to place it in the target user's hierarchy and
- to which degree ancestors of the new folder should also be created.

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.

4.3 Recommending Keywords

Meaningful annotations to bookmarks alleviate the retrieval of the bookmark when it is needed (see the recall problem from section 2) and help users to remember what a bookmark is about. This counteracts the effect of the lack of informativeness of the bookmark URL alone. For this purpose, the bookmarking facilities of most current web browsers provide input fields for descriptive content. Mozilla Firefox as an example provides input boxes for "Keywords" and a "Description". The bookmark system *SiteBar*, which serves as basis for the implementation of this approach, offers a single field "Description" - a minimal solution, but sensible for an architecture that aims at compatibility with a variety of web browsers.

Moreover, it is clear that a well-annotated set of bookmarks will serve as a much better foundation for any collaborative activity than just a set of plain URLs. Manual annotation however as an explicit interaction demands the

user's time and effort to come up with keywords and to type them in the appropriate input fields - a critical aspect for the adoption of a recommender system.

We propose to automatically suggest a limited number of high-quality keywords to a user when bookmarking a new URL. In this context, quality means that the keywords are specific and highly descriptive for the content of the website regarding the user's interest profile. The following factors should have an influence on the quality measure of keyword k_i for the link l:

- **number of users:**
  The more users have annotated l with k_i, the higher the quality.
- **similarity of users:**
  The more similar the users are that have annotated l with k_i, the higher the quality.
- **keyword frequency:**
  The higher the frequency of k_i in other users' descriptions of l, the higher the quality.
- **keyword specificity:**
  The more often a keyword is used for different links, the less specific it is; quality should decrease in this case. A popular mean for this purpose in the field of information retrieval is the TF-IDF weighting scheme. It defines the weight $w_{i,j}$ of term i in document j as

  $$w_{i,j} = tf_{i,j} \times log(\frac{N}{df_i})$$

  whereby $tf_{i,j}$ the frequency of term i in j, df_i is the number of documents containing i, and N is the number of all documents. Hereby, terms are penalized that are found across many documents.
  Applied to our scenario, one could notate the weight w_{l,k_i} of keyword k_i for link l as

  $$w_{l,k_i} = kf_{k_i,l} \times log(\frac{N}{N_{k_i}})$$

  whereby $kf_{k_i,l}$ is the frequency of keyword k_i in the description of link l, N is the number of all links in the system and N_{k_i} is the number of all links in the system annotated with keyword k_i.

Bringing it all together and normalizing it with the maximal quality value, the quality of keyword k_i for the link l newly bookmarked by user u is computed by:

$$w_{l,k_i} = \frac{\sum_{u_{rec} \in U_{l,k_i}} sim(u_{rec}, u) \times kf_{k_i,u_{rec},l} \times log(\frac{N}{N_{k_i}})}{max_{k \in K_{rec}} w_{l,k}}$$

whereby

- U_{l,k_i} set of users who have bookmarked l and assigned keyword k_i to it
- $kf_{k_i,u,l}$ normalized keyword frequency of keyword k_i in $keyw_{u,l}$
- $keyw_{u,l}$ set of keywords user u has assigned to link l
- N number of all links in the system
- N_{k_i} number of all links whose description contains k_i
- K_{rec} the set of all recommended keywords

The ten best-rated keywords are being recommended. If no other user has yet bookmarked the actual URL, the system looks for meta-information about keywords in the header of the website (`<meta name=''KEYWORDS''...`). If available, the system uses those keywords as default value of the description field.

Another assumption about a possible positive side-effect of recommending keywords is that it may support the unification of the annotation vocabulary used, especially inside user groups with similar interests. According to experience, even in such groups people eventually have different names for the same things. As an example, terms like *machine learning, data mining, pattern recognition* might be used synonymously. If the usage of one of the terms becomes prevalent in the group, its quality value will be higher at the time of the next keyword recommendation. This increases the probability that more users adopt this term. In the best case, this effect is self-enforcing, and leads by and by to a stepwise vocabulary unification. For the current state of the study, this remains speculation, but further research in this direction could be fruitful.

Besides the proposed collaborative approach, the next section presents two more possible approaches of bookmark classification for comparison reasons.

5. CONTENT-BASED APPROACHES

As detailed at the beginning of this section, several sources of information can be consulted to reason about automatic bookmark classification. For comparison reasons, two further algorithms were implemented: One is to try a classification based on the user's own classification history, a classical case of content-based recommendation. The last one is to consult a public directory, which is difficult to assign to a content-based or collaborative approach. It's not purely content-based as information other than only from the users themselves is used; but the collaborative idea of including similar users is also not matched.

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 Public Directory

The last alternative is to ask a public directory if and where it has classified the current website. As explained above, the public directories constitute strongly universal classification hierarchies for millions of websites and users. So the expectations to extract from them a classification tailored to a user's needs should not be too high. But if no other information is around, they are definitely worth to be

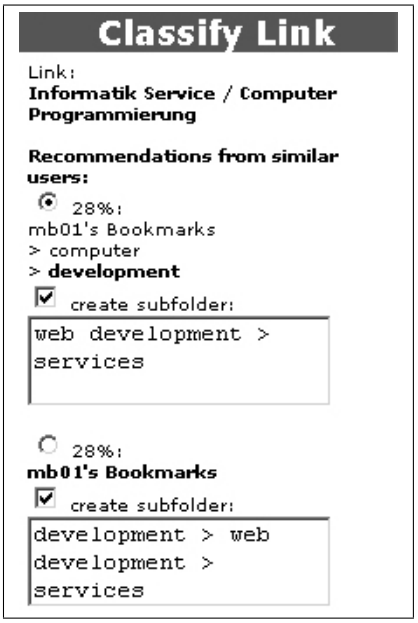


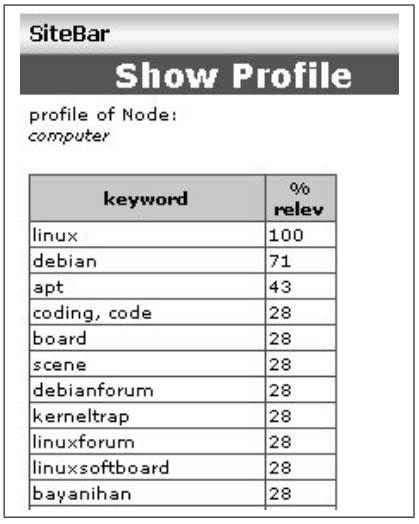
Figure 2: Screenshot of the user interface (Adding a new bookmark)

consulted, at least to enrich the current link with additional information.

The integration in the current system is done as follows: First, the public directory categories for the current URL are retrieved by sending a query to the Google directory, which searches the database of the open directory project [11]. The result page is parsed and the categories are extracted. Then a temporary bookmark folder with the category names as description is being created and its profile vector is initialized. For this folder, the system tries to find the most similar existing folder of the target user. If the similarity is above the threshold to create a new folder (see table 2), the matched folder is recommended, otherwise the system suggests to create a new folder inside the matched folder, namely the most specific one from the public hierarchy. If the system does not find a folder similar to the temporary folder, the recommendation is to create a hierarchy of the three most specific folders of the public directory in the user's root folder.

6. IMPLEMENTATION

The prototype implementation is called *CariBo* (**C**oll**a**bor**a**t**i**ve **Bo**okmark Classifier) and is based on the open source bookmark server *SiteBar* (*http://www.sitebar.org*). 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 2 shows the user interface where the outcome of the collaborative classification is presented to the user, Figure 3 depicts the display of a folder profile. Installation instructions and downloads can be found at our group website [4].



keyword	% relev
linux	100
debian	71
apt	43
coding, code	28
board	28
scene	28
debianforum	28
kerneltrap	28
linuxforum	28
linuxsoftboard	28
bayanihan	28

Figure 3: Screenshot of the user interface (Display a folder profile)

Nr. of test users	15
Average nr. of bookmarks	302
Average nr. of folders	45
Min / Max nr. of bookmarks	30 / 2662
Min / Max nr. of folders	9 / 269
Average nr. of bookmarks / folder	6,77
Total nr. of terms in database	12205

Table 3: Experimentation data statistics

7. EXPERIMENTAL RESULTS

A case study with 15 subjects, most of them students or staff of the Department of Computer Science of the University of Freiburg, was conducted. First, the subjects were asked to submit their existing bookmark file to the bookmark server. Table 3 summarizes some statistics about the experimentation data. Then, a set of 20 testlinks was generated by sequentially querying Google with each of 20 randomly picked terms from the database and adding the first search result (if not already present in the system) to the testlink set. To ensure some common bookmarks, all users were asked to bookmark all links contained in the testset and to file them manually to a folder of their choice. They were instructed to create a folder "unknown" for websites they were unable to classify. Before the testrun, the "unknown"-folder was removed from their collection.

For each user and each of his remaining testlinks, a "leave-one-out"-testing was applied: The current testlink was removed from his collection, and given to the three classification algorithms (collaborative, content-based, public directory). The outcome of each algorithm was a list with the top 5 recommendations of folders where the user could classify the bookmark. If the correct folder (i.e. the one where the testlink was taken out from) was among those, the classification was judged as a hit.

Figure 4 displays the results. The collaborative algorithm performed best and was in roughly 60% of all cases able to make a correct classification among the top 5 recommended folders. These results suggest that employing collaborative classification can outperform commonly used content-based classification approaches, whose hit rate was 38% in the ex-

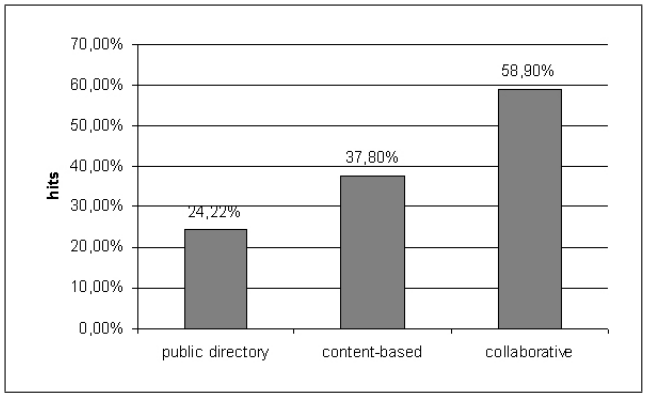


Figure 4: Experimental results

periment.

8. CONCLUSIONS AND OUTLOOK

This paper presented a novel approach to automatic bookmark classification, based on the classifications of similar users. The main methods presented were an algorithm to generate keyword recommendations for a user to annotate a new bookmark, and an application of a k-NN-classifier to generate collaborative recommendations for classifying new bookmarks. The latter uses a "augmented averaging"-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 experimentation, the collaborative classification outperformed clearly all other classification approaches. 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 - both of which were used for comparison only 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 / merging folders.

Long-term experimentations with direct user feedback on the classifications can be expected to further prove the utility of collaborative classification, especially with regard to the keyword recommendation. 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.

9. 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] 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.

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

[4] *CariBo - A Collaborative Bookmark Classifier*. http://www.informatik.uni-freiburg.de/cgnm/software/caribo.

[5] 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.

[6] 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 Access in Human-Computer Interaction (UAHCI)*, Las Vegas, Nevada USA, 22-27 July 2005.

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

[8] 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.

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

[10] *Onix Text Retrieval Toolkit - freely available stopword list.* http://www.lextek.com/manuals/onix/stopwords1.html.

[11] *Open Directory Project*. http://www.dmoz.org.

[12] 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.

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

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