Towards Effective Browsing of Large Scale Social Annotations

Rui Li[1*], Shenghua Bao[1*], Ben Fei[2], Zhong Su[2], and Yong Yu[1]
[1]Shanghai JiaoTong University
Shanghai, 200030, China
{rli, shhbao, yyu}@apex.sjtu.edu.cn
[2]IBM China Research Lab
Beijing, 100094, China
{feiben, suzhong}@cn.ibm.com

ABSTRACT
This paper is concerned with the problem of browsing social annotations. Today, a lot of services (e.g., Del.icio.us, Filckr) have been provided for helping users to manage and share their favorite URLs and photos based on social annotations. Due to the exponential increasing of the social annotations, more and more users, however, are facing the problem how to effectively find desired resources from large annotation data. Existing methods such as tag cloud and annotation matching work well only on small annotation sets. Thus, an effective approach for browsing large scale annotation sets and the associated resources is in great demand by both ordinary users and service providers. In this paper, we propose a novel algorithm, namely Effective Large Scale Annotation Browser (ELSABer), to browse large-scale social annotation data. ELSABer helps the users browse huge number of annotations in a semantic, hierarchical and efficient way. More specifically, ELSABer has the following features: 1) the semantic relations between annotations are explored for browsing of similar resources; 2) the hierarchical relations between annotations are constructed for browsing in a top-down fashion; 3) the distribution of social annotations is studied for efficient browsing. By incorporating the personal and time information, ELSABer can be further extended for personalized and time-related browsing. A prototype system is implemented and shows promising results.

Categories and Subject Descriptors
H.3.3 [Information Search and Retrieval]: Information filtering

General Terms: Algorithms, Experimentation, Human Factors

Keywords: Annotation Browsing, Clustering, Evaluation, Social Annotation.

1. INTRODUCTION
With the rapid development of new technologies, both ordinary users and service providers are experiencing the coming wave of the next-generation Web. As a representative, tagging based websites, such as Del.icio.us[1] and Flicker[2], have achieved a significant success. Their low technical barriers and the easy use of annotations have attracted lots of users. Millions of annotations are freely and openly assigned to the digital items like web pages, photo images and blog posts. Now, annotation is not only a method for organizing contents to facilitate the users who create it, but also a navigation mechanism for users to discover interesting resources. It has become a new interface of Web and has drawn much attention from both research and industrial communities.

Currently, there are two main methods of helping users to seek the information through annotations. The first one is the keyword-based search, which is the most common way for finding information on the Web. Systems of this type will display all contents associated with the given annotation. The second one is a method called tag cloud view [20]. It usually displays the social annotations alphabetically with different font sizes and colors indicating their popularities. Selecting a specific annotation will generally lead to the keyword search with the selected annotation as input. Compared with the direct searching method, tag cloud provides a better user interface for browsing the popular social annotations. However, the drawbacks of the methods are obvious, especially when the scale of social annotations is quite large:

- Contents and annotations are simply matched by the literal representations instead of the semantic meanings. The synonymy and ambiguity problems exist in these systems. The synonymy problem arises when different annotations have similar meanings. As a result, simple matching can not find all the relevant resources. The ambiguity problem arises when an annotation has several different meanings. It will introduce noise pages. Both of these problems influence the performance of existing browsing systems.
- The returned results are not categorized. It is difficult for users to browse thousands of returned results to find their interested resources.

In this paper, we consider the problem of browsing large scale social annotation data. An effective algorithm, Effective Large Scale Annotation Browser (ELSABer) is proposed. Compared with the previous method, it has the following advantages:

1. **Semantic Browsing:** We give the measurement of semantic similarity among annotations. Each annotation is assigned with a semantic concept consisting of the semantically related annotations, thus the synonym problem can be controlled by matching the semantic concept of the selected annotation. For example, assuming one selects the annotation "book", the resources annotated by either "book" or "books" will be returned since "books" also appears in the concept of "book".

---

[*] Part of Rui Li and Shenghua Bao's work of this paper was conducted in IBM China Research Lab.
[1] http://del.icio.us
[2] http://www.flickr.com

Copyright is held by the International World Wide Web Conference Committee (IW3C2). Distribution of these papers is limited to classroom use, and personal use by others.
WWW 2007, May 8–12, 2007, Banff, Alberta, Canada.
ACM 978-1-59593-654-7/07/0005.

2. **Hierarchical Browsing**: Users can locate their desired resources more easily if social annotations are organized hierarchically, especially for large scale annotation sets. When annotations are organized in this way, the click trace of a user represents his intended target in a top-down fashion without ambiguity. We observe that the social annotations present concepts in different semantic levels and we build a hierarchical structure for social annotations based on this observation.
3. **Efficient Browsing:** The time cost for browsing increases with the growth of the size of social annotations. In ELSABer, a sampling method based on the power low distribution of annotations is applied for efficient browsing.

Given the personal information and the time-related restrictions, ELSABer can be easily enhanced. By incorporating users' profiles, ELSABer can display the social annotations and contents according to users' preferences. By further incorporating the time restrictions, ELSABer can display the popular annotations and contents within a specific time interval. These two extensions are useful in helping the user to discover the personal interested resources and the popular resources.

Figure 1. Interface of Social Browsing System

A prototype system is implemented based on ELSABer. Figure 1 gives a snapshot of the system. The page behind is the initial interface of the system. It contains popular annotations distributed in different clusters. The size of each annotation indicates its popularity. The page in the front is the result after a user selects the annotation "programming". On the right side is a set of pages related to the current annotation. Each line on the left side is a sub-category of the current annotation, which consists of several related annotations. Users can click the tag on the left side to further investigate that category. In this paper, we use the terms "annotation" and "tag" interchangeably.

The rest of the paper is organized as follows. Section 2 briefly reviews the studies of social annotation. Section 3 gives an overview of our algorithm. Section 4 describes each component in our social browsing algorithm. Section 5 discusses how to extend the algorithm in two ways. Section 6 presents the experimental results. Finally we make a conclusion in Section 7.

2. RELATED WORK

2.1 Research on Social Annotation

Recently, there are lots of studies on social annotations, including blog posts, interesting demos and academic research papers.

The early discussion of the social annotation can be found in [1, 3, 4, 5, 11]. They initiated the idea that sharing tags can lead to the concept known as "folksonomy". The term first appeared in an information architecture mailing list [12]. Quintarelli [5] suggested that we should take social annotation as an information organizing tool. In [1], Golder et al. gave the specific analysis of the social annotation data in Del.icio.us in both the static and dynamic aspects. In [3], the authors gave a good review of available social bookmark tools.

Research based on social annotation has been done in various areas such as semantic web [6], social network [9], and enterprise search [8]. In [9] Mika proposed a bipartite model of ontology with a social dimension and found that the semantic relationships among tags are based on their co-occurrence with users or resources. In [6], Wu *et al.* used a probabilistic generative model to obtain the emergent semantics hidden behind the co-occurrence of three types of data (tags, users and resources). They also proposed a framework for semantic search based on their emergent semantic model. In [8], Dmitriev suggested that folksonomies were not limited to the blog sphere but also benefited enterprise search. An annotation tool was implemented within an enterprise environment and improved the search efficiency. In [7], the authors analyzed the effectiveness of tags for classifying blog entries and argued that there is a topical hierarchy among tags. However, the hierarchy, which is a static rigid binary tree lack of semantic control, is not suitable for the social browsing problem.

All the above research is different from ours. Their work is about discovering and utilizing the features of social annotations instead of focusing on how to browse annotations themselves.

2.2 Social Browsing and Navigation

There are a few studies on visualizing and browsing the social annotations. Dubinko [10] proposed and solved the problem of visualizing the evolution of tags within Flickr online image sharing service. They gave an efficient algorithm for processing the large data in real time. Their work focuses on discovering the hot images and tags in a pre-defined time interval. It is not a proper solution for users to browse all annotations. In [21], Begelman applied the clustering algorithm on the social annotations to improve users' browsing experience. Their algorithm can not handle the synonymy and ambiguity problems. Our work is different from theirs. We proposed the browsing framework with three features, including the solution for a semantic browsing.

Some demos for visualizing tags are also available on Web. Grafolicious [22] produces graphs illustrating when and how many times a URL has been bookmarked in Del.icio.us. HubLog [23] gives a graph of related tags connected with the given tags. Although these demos gave a vivid picture of social annotations in different aspects, their goals are not to help users to browse annotations effectively.

3. ELSABER OVERVIEW

In this section, we give an overview of the ELSABer algorithm as shown in Algorithm 1. The algorithm is generally designed for any social annotation environment, e.g., Del.icio.us, Flickr. In this paper, we use Del.icio.us annotations for analysis and evaluation.

From step 1-1 to 1-2, the algorithm initializes the first view of annotations. N_T, N_U, N_C, and N_{CT} denote the number of tags, URLs, clusters and tags in each cluster. In our experiment, these parameters are set to 2000, 2000, 20, and 5, respectively, which means the top 100 tags distributed in 20 clusters on 2000 most frequent tags and URLs are presented to the users as the default

browsing interface. These popular tags, which are associated with a large number of resources, are selected as the roots in hierarchical browsing. When users select a tag as the entrance to annotation browsing, the algorithm outputs its related resources and a set of annotations as sub-tags. Users can iteratively select any annotation from the displayed sub-tags for further exploration. The iterative process consists of four components as follows:

- Tag selection (from 2-1 to 2-3): to provide a semantic browsing, the algorithm takes the selected tag as a semantic concept which consists of several highly related tags. The user's path from the root to the current annotation forms a set of concepts and specifies the user's interests. The URLs and tags related with the concept set will be selected.
- URL and tag sampling (step 3): this is an optional step. We introduce a sample mechanism to sample tags and URLs which match the specified concept set. The application of sampling assures that the algorithm is always running on a data set with controlled size.
- Sub-Tag Generation (step 4-1): we develop a set of features and rules to find the sub-tags of the current tag. The resources of the current tag can be further classified into concepts of these sub-tags.
- Similarity based Clustering (step 4-2): a social clustering algorithm is introduced to find a proper number of clusters from the generated sub-tags in the previous step. Then the sub-tags in each cluster are presented to the user.

Then the user can click one of these presented sub-tags to further seek his desired resources.

Algorithm 1 Social Browsing Algorithm

Input	An empty concept set S_C
Step 1	Output the initial view of annotations
1-1	Select top N_T tags S_T and top N_U URLs S_U
1-2	*Social Clustering* (S_T, S_U, N_C),
Return	N_C clusters C_T with top N_{CT} tags in each cluster
Loop	User select a tag T_i
Step 2	Concept Matching
2-1	Calculate related tags to T_i to construct concept C_i
2-2	Add C_i to S_C
2-3	Select *related* URL set SU_i and *related* tag set ST_i
Step3	Get *sample* URL set SU_s' and *sample* Tag set ST_s'
Step4	Hierarchical Browsing
4-1	Calculate Sub-tag set SST_i w.r.t. concept C_i
4-2	*Social Clustering* (SST_i, SU, N_C), obtained CT_j
	Return top N_{CT} tags in each cluster
IF	Termination condition Satisfied; **Return**
ELSE	**Loop**

4. BROWSING SOCIAL ANNOTATIONS

In this section, we will give a detailed description of each component in ELSABer. Before the discussion, we first give a formal representation of annotation data.

4.1 Data setup and representation

Del.icio.us provides a popular tool for organizing bookmarks. A description online [2] states it as:

"*A social bookmark manager. It allows you to easily add sites you like to your personal collection of links, to categorize those sites with keywords, and to share your collection not only between your own browsers and machines, but also with others*".

In Del.icio.us, an annotation activity typically consists of four elements: an annotator, a URL, a tag, and a tagging time. We define an annotation as a quadruple:

(User, URL, Tag, Time).

We disregard the roles of User and Time, and view the annotation data as points in a high dimensional space called the tagging space. The annotation data can be represented as an $m \times n$ matrix C, where m and n is the total number of tags and URLs, respectively. Let C_{ij} denote the number of *users* who annotate the jth URL with the ith tag. Let M be the $m \times n$ association matrix and M_{ij} denote the association degree between the ith tag and the jth URL. A simple method is to let M_{ij} equal to C_{ij}. In our experiment, we borrow the idea of *TFIDF* from the IR field and calculate the association weight as follows:

$$M_{ij} = C_{ij} * \log\left(\frac{n}{|URL(t_i)|}\right), \quad (1)$$

where $|URL(t_i)|$ represents the number of URLs annotated by t_i.

Given the matrix M, the tag can be represented as a row vector T_i (U_1,U_2,.. U_n) of M. Similarly, the URL can be represented as a column vector U_i (t_1,t_2,…,t_m) of M.

4.2 Semantic Browsing

4.2.1 Annotation Similarity Estimation

People annotate web pages mainly for organizing web pages with different contents, so annotations are usually abstracts of the contents of these web pages. Abstracts of the same web page are usually similar. Below, we give the first observation of social annotations:

> **Observation 1**: *similar tags will annotate similar URLs and similar URLs will be annotated by similar tags in the social annotation environment.*

Based on observation 1, the semantics of a tag can be reflected by resources which it tagged. The semantic relation is derived from their co-occurrences. As shown in Figure 2, T_1 and T_2 are similar tags since they share similar URLs U_3 and U_4. T_1 and T_3 are less relative. Figure 2 also illustrates that the similar tags also annotate the different URLs.

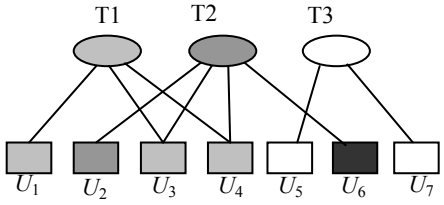

Figure 2. Similar Annotations

For measuring the semantic relationship between tags, we propose a symmetric measurement as follows:

$$Sim(t_i, t_j) = \cos(T_i, T_j), \quad (2)$$

where T_i and T_j are tag vectors corresponding to tags t_i and t_j, respectively. The tag vector T_i is determined by the URLs which are annotated by the tag. So it may vary according the change of the related URLs.

Some linguistic features are also used for calculating $Sim(t_i, t_j)$. When tags are freely assigned to the relative URL, tags are used in various forms, such as the plural form and gerundial form. For example, "Programs", "Programming" and "Program" all exist in the annotation data. Additional weight is added to $Sim(t_i,t_j)$ if two terms share the same etyma after porter stemming. Besides, if the two terms share the etyma after eliminating the external punctuations, a lighter additional weight is added to the $Sim(t_i,t_j)$ score. The two weights are set to 0.1 and 0.08, respectively.

4.2.2 Generating the Semantic Concept

In the social browsing setting, the user tries to find his desired resources by selecting the tag with the closest meaning to his intended information. Therefore, when the user selects "film", those pages tagged by "movies" are also of his interest. In order to provide the user with the complete resources of his interest, we find tags and URLs related to the semantic concept of a tag instead of finding tags and URLs by matching the tag literally. Following, we give the method of generating semantic concepts.

Given the selected tag t_i, we choose a set of tags most related to t_i, as the synonymic tag set $ST_i = \{ t_j \mid t_j \text{ is similar to } t_i \}$. In this paper, the candidate tags ST_i is generated using the following rules:

1. t_j should be among the N most similar tags related to t_i
2. The similarity should be larger than a threshold θ.

where N and θ in the rules are set to 4 and 0.7, respectively. Then, a semantic concept C_i for t_i is represented by the following set:

$$C_i = ST_i \cup \{t_i\}, \quad (3)$$

Note that, once a user has clicked L times and forms a click trace of t_1, t_2,…,t_L, we have a sequence of L concepts: C_1, C_2,…,C_L. Let $S_C = \{C_1, C_2, \ldots, C_L\}$. The related URLs in step L can be defined as follows:

$$\mathrm{Re}URL(S_C) = \{u \mid \forall C \in S_C, T(u) \cap C \neq \Phi\}, \quad (4)$$

where u is a candidate URL and $T(u)$ means the set of annotations given to URL u.

Given a set of related URLs, the related tags can be defined as all the tags given to $ReURL(S_C)$.

$$\mathrm{Re}Tag(S_C) = \{t \mid \exists u \in \mathrm{Re}URL(S_C), t \in T(u)\}, \quad (5)$$

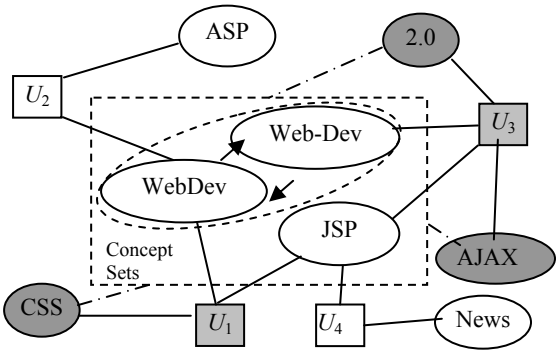

Figure 3. Matching with Concept Set

Figure 3 illustrates the idea above. The concept set contains two concepts: one consists of "WebDev" and "Web-dev" while the other contains tag "JSP" only. "WebDev" and "Web-Dev" are highly related tags which satisfy the above rules. U_1, U_3 are the URLs matching the current concept set since U_1 is annotated by "WebDev" and "JSP" while U_2 is annotated by "Web-Dev" and "JSP". Related tags like "CSS", "AJAX", "2.0" are obtained from the matched URLs. The sub tagging space is formed by these related tags and URLs.

Note that, by tracking the user's selections, the problem of tag ambiguity can be solved, because previous selected concepts play as domain limitations, which can disambiguate meanings of tags in different domain. For example, U_4 will not be selected, since it does not match the limitation "Web-Dev".

4.3 Hierarchical Browsing

4.3.1 Hierarchical Structure

Quintarelli [5] and Mathes [4] both argue that the tagging space is a flat space and a hierarchical representation of topics does not reflect the associative nature of social annotations. But in [1], Golder states that the different expertise and purpose of tagging participants may result in tags at various levels of abstraction to describe a resource. For example, a photo can be tagged at the "basic level" by "cat", at a super ordinate level by "animal" or at various subordinate levels below the basic level by "Persian cat" or "Felis silvestris cats longhair Persian". Our observation is that:

Observation 2: *there is not a neat tree structure like taxonomy or human built ontology with rigid hierarchies and pre-defined categories with clear boundaries for social tags, but the tags used in social annotation do locate in different semantic levels in the social annotation space.*

By our observation there are many combined words like "programming/java" and "Design/CSS", which may reflect the needs for hierarchical annotation in Del.icio.us. Several single word tags which are used to annotate the URL by the same user also reflect that the hierarchy exists in the social annotation. For example, there are URLs tagged by "java, jdbc" and "music, jazz". So it's feasible to explore the hierarchical structure of social annotations for hierarchical browsing. The structure has the following features:

- There are multiple ways to the target resource. To find pages about jsp we can make the clicks of "programming", "java" and "jsp" in order or we can follow the path from "design", "web" to "jsp".
- The categories are not rigid. One URL can be associated with several categories, since different people have different views about the same thing.
- The hierarchical structure is dynamic with the increasing of social annotation data. If resources associated with a leaf tag increase, the leaf tag will extend several children nodes.

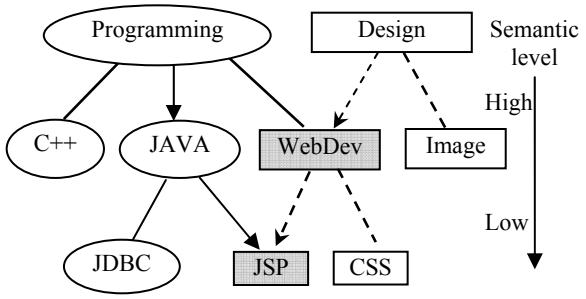

Figure 4. Hierarchical structure

Figure 4 illustrates the features described above. There are two hierarchical structures, rooted "programming" and "Design", respectively. The annotations like "WebDev" and "JSP" are shared in both trees. Users can reach the "jsp" tag from either "programming" or "Design".

4.3.2 Sub-Tag Generation

For each selected tag, a set of related tags are obtained from previous steps. Obviously, not all the tags are proper to be the child node of the current tag to expand the tree structure. Related tags are mainly of the following types:

1) Father Tags: These tags are in a higher semantic level, which have more general meanings than the current tag/ node.

2) Sub Tags: These tags are in a lower semantic level. A tag of this type describes the content of the related URL more specifically, or is a sub-category of the current node. These tags are likely to be good candidates for expanding our social browsing tree structure.

3) Other Tags: Tags of this type describe different resources related with the current tag, but they are neither more general nor more specific than the current tag, so they are useless for hierarchical browsing.

Several features are explored to identify whether a tag is a sub-tag of the current tag/node. In the following, t_i denotes the candidate sub-tag and t_j denotes the current tag.

1) **Coverage of Tags**: Tags in different semantic levels have different content coverage. A tag in a high semantic level covers more resources than its sub-tags. For example, there are more URLs associated with "Web" than "Google" and more with "Google" than "Googletips". The coverage relation between two tags is defined as:

$$Coverage_{ij} = \frac{U(t_i)}{U(t_j)}, \tag{6}$$

where $U(t_i)$ denotes the number of URLs tagged with t_i. Figure 5 illustrates the capacity of this feature. Here, the current tag is "Google" and the Coverage values of 40 of its related tags are calculated and compared. The types of these tags are manually labeled. The figure shows that tags at a super ordinate level such as "Web" and "computer" have much higher Coverage values than other tags.

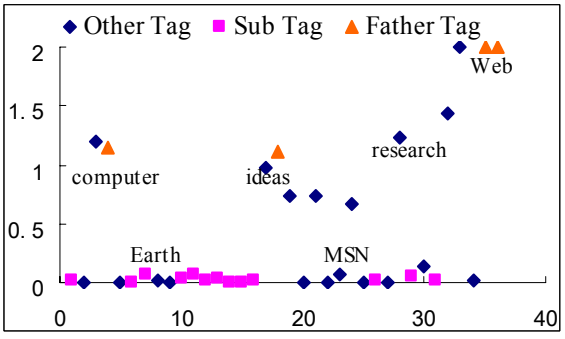

Figure 5. Identify sub-tags using coverage

2) **Intersection Rate**: This feature, denoted by IR, is defined as follows:

$$IR_{ij} = \frac{U(t_i) \cap U(t_j)}{U(t_i)} \tag{7}$$

It is the ratio of the number of t_i and t_j's common URLs to the number of t_i's URLs. If the intersection URL set is the main part of all the URLs of t_i, but a small part of t_j, we can infer that t_i is a sub-tag of t_j. For example, most URLs tagged by "gmaps" are also tagged by "Google", but a small number of URLs tagged by "Google" are tagged by "gmaps", so we infer that "gmaps" is a sub-tag of "Google".

Note that we should pay attention to the size of $U(t_i)$, if $U(t_i)$ is as small as 1 or 2, the IR_{ij} above will give the tag t_i a high score and take it as a sub-tag with high confidence. But the tag is likely to be meaningless tag or noisy tag. We use equation 8 for solving this problem; we set a threshold on the size of $U(t_i)$. We also introduce a parameter λ for smoothing the results. In the experiment, both the threshold and λ are set to 5.

$$IR_{ij} = \frac{U(t_i) \cap U(t_j)}{U(t_i) + \lambda} \tag{8}$$

Figure 6 shows the capacity of this feature. The current tag and the data set are same as those in Figure 5. We can see from Figure 6 that tags at subordinate levels like "gmaps" and "earth" have much higher IR values than other tags.

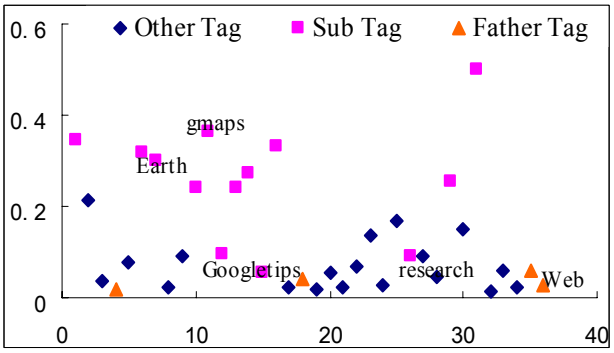

Figure 6. Identify sub-tags using intersection rate

3) **Other Features**: We also use some features as the supplementary for identifying the sub-tag relationship. The third feature is called Inverse-Coverage Rate, which is defined by Equation 9. And Feature 4, namely *IR*' is defined by Equation 10. Feature 5, *IRR*, is set to a set of discrete values of 1, 2 and 3 according to their rank by *IR*. If a tag ranks top 30 among all the tags, its *IRR* value is set to 1, if it ranks between 30th and 60th, its *IRR* value is 2, otherwise it will have an *IRR* value of 3.

$$ICR_{ij} = \frac{U(t_i) - U(t_i) \cap U(t_j)}{U(t_j)}, \tag{9}$$

$$IR'_{ij} = \frac{U(t_i) \cap U(t_j)}{U(t_j)} \tag{10}$$

Given the features above, each related tag is represented as a feature vector. A decision tree can be derived from the manually labeled data set to predict the sub-tag relations using C4.5. Figure 7 shows a snippet of the rules we got.

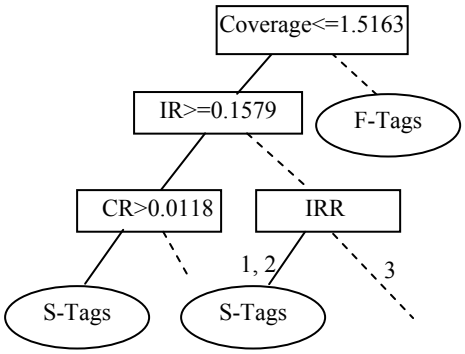

Figure 7. Rules for identifying sub-tag

4.3.3 Sub-Annotation Clustering

For categorizing the sub-annotations generated in the last step, we present a clustering algorithm, which successfully solve the following problems:

1) **Whether the current tag needs further expanding:** If the number of related URLs of the current tag is less than a predefined threshold, there is no need to further expand it. In our experiment, the threshold is set to 20.

2) **How many clusters to expand to:** The number of clusters after expansion is quite an influential factor to the browsing experience of the users besides its influence on the quality of the clusters: too few clusters will cause low cohesion within each cluster and too many clusters will frustrate the user, and we can not fix the structure of the browsing tree beforehand, i.e., we can not set the tree to be a binary tree or a K-tree.

In the light of the above discussion, traditional clustering algorithms like K-means [14], which purely rely on a predefined cluster number is not proper for our problem. Clustering algorithms based on graph partition [13] will give an optimal partition of the graph, but complexity of these algorithms prohibits them to be applied in a real-time browsing problem. Here we propose a dynamic clustering algorithm for the social browsing problem, as shown in Algorithm 2. It may not be the optimal solution according to the graph theory, but it is a proper solution for social browsing. The label of each cluster is also generated during the clustering process.

Algorithm 2 Social Clustering Algorithm

Input	Tag Set T_S , URL Set U_S, max number of clusters N_C
1	Get all the candidate sub-tags and rank them by the informative rank.
Loop	Select tag T_i which has the top informative rank
2-1	Get a list of tags Ts' with similarity threshold 0.4 from T_S and add them to C_i. Remove tags in T_S' from T_S
2-2	Get a list U_S' of URLs similar with the tags in C_i. Remove URLs in U_S' from U_S
If.	(Size of U_S==0) or (number of clusters ==N_C) **Return**
Else	**LOOP**

The algorithm first introduces an informative rank over the candidate sub-tags based on the following tag properties:

Tag Frequency/ Inverted URL Frequency: This property indicates the annotation's importance and is defined in the same way as Equation (1).

Intra-Cluster Similarity: This property, namely *ICS* was used to measure whether a tag is a good representation of a single topic.

$$ICS = \frac{1}{U(t)} \sum_{u(i) \in U(t)} \cos(u_i, o_t) , \quad (11)$$

Where o_t denotes the centroid of all the URLs associated with the tag and each URL in this tagging space can be represented as a vector $u_i=(t_1,t_2,\ldots t_n)$ (see section 4.1):

$$O_t = \frac{1}{U(t)} \sum_{u_i \in U(t)} u_i , \quad (12)$$

where $U(t)$ represents the number of URLs associated with tag t.

Tag Entropy: This property, denoted by *TE*, is used to calculate the distinctness of a tag [15]. A tag which seldom shares URLs with other tags is more likely to be a cluster.

$$TE(t) = \sum_{t_i} \frac{U(t) \cap U(t_i)}{U(t)} \log \frac{U(t) \cap U(t_i)}{U(t)} , \quad (13)$$

Finally, the informative score for tag t is defined as the linear combination of all the above properties:

$$Infor(t) = w_1 TFIDF(t) + w_2 ICS(t) + w_3 TE(t) , \quad (14)$$

We decide the weights w_1, w_2 and w_3 by using a linear regression model over the manually labeled data set. In our experiment, these weights are 0.58, 0.27, and 0.13, respectively.

After obtaining each tag's informative score, we select the most informative tag as the label of the first cluster, and find its similar tags using Equation 2 and its related URLs by calculating the *cosine* similarity between these URLs and the centroid of all the similar tags. Then we remove its similar tags and related URLs from our dataset. This process terminates when no remaining tag has enough number of related URLs.

4.4 Efficient Browsing

4.4.1 Observations

Nowadays the number of tags and URLs are increasing exponentially with the development of Web 2.0 and extensive application of tagging services. Therefore, the efficiency of our algorithm will be influenced. In this section, we will discuss how to accelerate our algorithm. First we give the 3rd observation based on our analysis on Del.icio.us.

Observation 3: *Popular tags and URLs play an important role in our social annotation data. People use popular tags to annotate URLs and also the popular URLs are annotated by the majority of tags.*

Figure 8 and Figure 9 demonstrate Observation 3 illustratively. X axis represents tags in the order of their counts and Y axis represents the counts of the tags. Figure 8 illustrates the distribution of the counts of tags which are associated with a certain URL. We discover that people always use most popular tags to annotate the URL and unpopular tags are barely applied to annotate. Figure 9 demonstrates the distribution of the counts of tags associated with the whole Del.icio.us data. We find out that the popular tags are frequently and extensively used in the whole Del.icio.us data although there are thousands of tags used.

4.4.2 Acceleration using Sampling

The responding time of our browsing algorithm is the key to the users' experience. For the sake of efficiency, we borrow the idea of the inverted index from the IR area to index both the tag vector and the URL vector. However, direct application of this indexing

scheme would still be inefficient because the tagging space has billons of annotations. To overcome this difficulty, we introduce a sampling method based on Observation 3 to limit the time complexity of our algorithm to a proper scale in spite of the huge size of social annotation data.

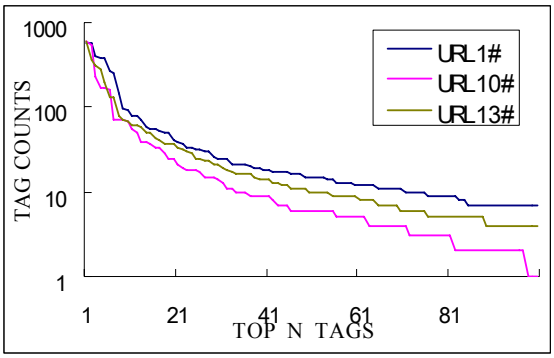

Figure 8 . Tag distribution over URL

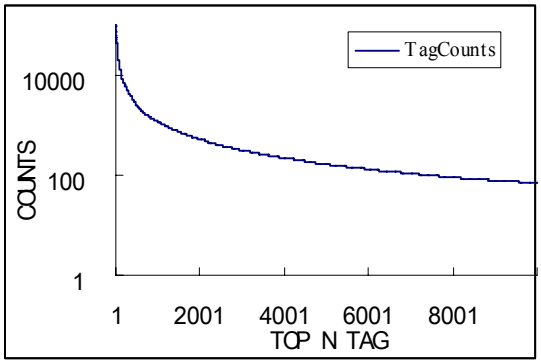

Figure 9 . Tag distribution over Tag Counts

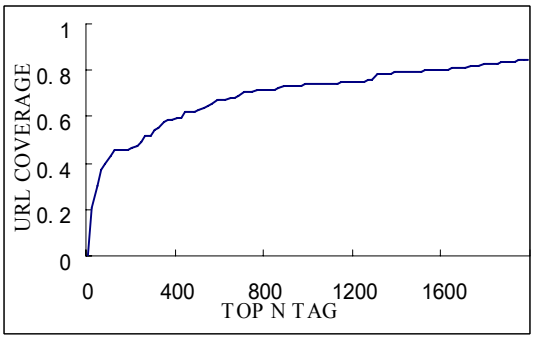

Figure 10. Illustration of Tag-URL coverage

According to Observation 3, we discover that the content of a URL can be reflected by the most popular tags and also the tag semantics can be decided by the most popular URLs. So we can get good results efficiently by running our algorithm in a small sub tagging space consisting of tags and URLs that are most frequently used and annotated, i.e., sampling *K* most frequently annotated URLs and *K* most frequently used tags from the dataset to form a sub tagging space for the algorithm. In our experiment, we set *K* to 2000, so the size of *M* is 2000 × 2000.

Note that we do not cut off the "long tail" of tagging space, although we use the sampling method. In our algorithm, the more important a tag is, the earlier it will emerge. Based on the discussion above, these important tags cover the majority of URLs, thus should be located at higher semantic level and presented to the user earlier. After the user click one of these tags, related URLs associated with this tag will be discovered, which will bring related tags including both popular and unpopular ones. Therefore we do not lose the connection with the "long tail". After a sequence of click by the user, the intention of the user will be more specific, this causes a decreasing number of related URLs or related tags. When the number is less than 2000, all the tags and URLs will be calculated, which means that the "long tail" is covered and the sampling method is not applied. Figure 11 shows the URL coverage of popular tags in Del.icio.us. Axis X is the *ith* popular tag, and Axis Y is the URL coverage percentage. It shows 84% of URLs in our data set associate with the top 2000 tags.

5. ENHANCED MODELS

Different people have different browsing preferences, since users have different interests. The personalized browsing provides the user with annotations more closely match his interests by utilizing user preferences. People are also interested in recent hot topics. The time-related browsing can discover annotations according to their popularity within a specific time interval. In this section, we will show that our browsing framework can be easily extended to fit the requirement of personalized and time-related browsing.

5.1 Personalized Social Browsing

Personalized browsing has been well studied in browsing interface [16], personalized website browsing [17], personalized webpage recommendation [18], etc. Here, we are to provide the personalized social annotation browsing.

In previous personalized systems, additional effort is usually required to build a user profile which is generally time consuming, and the generated profiles are sometimes out of date. In our social annotation environment, the profile of a user can be directly obtained from Del.icio.us and dynamically modified according to the changes of the user's interests over time. Assuming that User U_a is a registered user of Del.icio.us, his profile is represented as a set of triples:

$$P(U_a) = \{ (\text{User}, \text{URL}, \text{Tag}, \text{Time}) \mid \text{User} = U_a \}, \quad (15)$$

Given the profile $P(U_a)$, the social annotations can be classified into three categories as shown in Figure 11:

1) User Annotated: This category contains tags and resources that appear in $P(U_a)$ (The black ones in Figure 11)
2) User interested: This category contains tags and resources that are related to $P(U_a)$. They are the annotations that the user may be interested in (The gray ones in Figure 11).
3) Not interested: This category contains tags and resources that have no/few relations with $P(U_a)$. (The blank ones in Figure 11).

The user interested annotations and resources can be found as follows:

$$UI(R_i \mid P(U)) = \max_{R_j \in P(U)} \cos(R_i, R_j)$$
$$UI(T_i \mid P(U)) = \max_{A_j \in P(U)} \cos(T_i, T_j), \quad (16)$$

where $UI(R_i|P(U))$ denotes the degree of interest between user U and resource R_i, while $UI(T_i|(P(U))$ denotes the degree of interest between user U and annotation T_i. R_i denotes the vector

representation of a resource, and T_i denotes the vector representation of A_i.

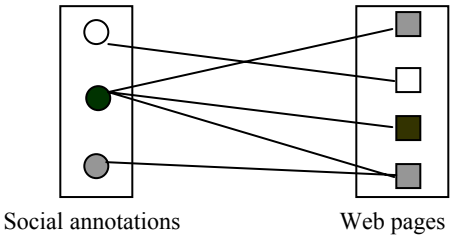

Figure 11 User-interested illusion

Given this quantitative evaluation of user interests of each annotation and the resource, we extend the basic social browsing model to the personalized model as follows:

1. Distinguish the user interested resources from the rest. The user interested annotations found by Equation 16 may help the user a lot in finding the unexpected but potentially interesting resources.
2. Adjust the sampling and ranking algorithms according to the user's preference. We can incorporate the user's preference into the ranking algorithm in section 4.3 as an additional property. Then the informative score is calculated as:

$$Infor(t,U) = \alpha \times Infor(t) + \beta \times UI(t \mid P(U)) \quad (17)$$

5.2 Time Related Browsing

As described in [19], the current Web is a sensor of the real world in some sense. In most cases, users are often interested in browsing the popular web pages in recent time. Thus providing time-related browsing would be helpful to most users. Recall that the annotation data in Del.icio.us can be represented as a set of quadruples:

(User, URL, Tag, Time)

Where "Time" is the time when "User" tagged "URL" using tag "Tag". Due to the fact that different users may annotate the same URL at different times, the times related to a specific URL form a time sequence TS [t_1, t_2,… t_n]. Given the user required time interval TI= [t_s, t_e]. We define the match of the URL's time sequence TS and the user required time interval TI as follows:

$$Match(TS,TI) = \begin{cases} true & if \ \frac{|t_e - t_s|}{|t_e - t_1|} >= \theta \\ false & else \end{cases}, \quad (18)$$

Where t_e-t_s denote the number of URLs annotated in this time interval. Then we extend the basic model to the time-related social browsing model by applying the social browsing algorithm only on the matched tags and their associated URLs. Here θ is set to 50%.

6. EXPERIMENTAL RESULTS

In this section, we give the evaluation of the proposed algorithm. All the discussions below are based on a data set collected from Del.icio.us during May, 2006, which consists of 1,736,268 web pages and 269,566 different annotations.

6.1 Similarity measurement

For evaluating the effectiveness of our similarity measurement, we give several results of constructing a concept for a given tag. As shown in Table 1, the concept of a given tag consists of several highly related tags that are generally synonymous tags, abbreviations or plurality of the given tag. The semantic of the current tag is influenced by the user's previous selections, e.g., "movie" in the "programming" area may mean "screenshot" or "screen capture" instead of "film" or "moving picture" in daily life; the concept of "Brainstorming" in the programming area are more likely to be "mindmap" and "freemind", which are two popular software used for brainstorming. Results show that our similarity measurement can correctly reflect relations between tags in the social annotation environment.

Table 1. Results of Synonym Tags

CVS : Versioncontrol, SVN, subversion, control
Movie: Movies, Film ,Films
Computer/Gallery: album photogallery fotos
Programming/Meta: Metaprgramming
Programming/Movie: screenshot screencapture
Programming/Brainstorming: mindmap freemind

6.2 The results of clustering

Table 2 shows the sub-tags discovered for 20 selected concepts. These concepts are selected randomly from 100 popular topics of Del.icio.us by a group of students in our lab. The concepts are distributed in different areas. Since a lot of annotations in Del.icio.us are related with IT, more selected topics are related to IT. In each box, the first line gives the concept label. For each of the 4 concepts in the first line, we listed only ten subordinate concepts and for each of the rest of concepts, we listed only 5 subordinate concepts due to space limitation. Table 2 implies that our algorithm is able to organize a hierarchical structure of tags as people think in their everyday life. For example, when the user clicks *science*, our algorithm is able to generate a series of sub categories such as *math*, *physics*, *psychology*, etc., which are mostly meaningful and very distinguishable, and also illustrate the topic most efficiently according to the knowledge of people.

6.3 Execution time

We evaluate the efficiency of our system with a modest machine (Intel Pentium IV 3.0 GHz, 1GB memory, 2 processors). The system is implemented in the java language. Lucene API is also used to build URL and Tag index. We cached the matrix of top 2000 URLs and 2000 tags for reducing the time cost of database accesses. The average time of processing 20 concepts is 1.3 sec.

Figure 12 User-interested tags

6.4 Case Study

6.4.1 Personalized Browsing,

Figure 12 shows the results of the algorithm with the consideration of the user's profile. The tags in red are owned by the user and the tags in orange are recommended tags for the user.

In the experiment, the user's profile consists of 25 tags and 45 URLs and the top 5 user interested tags are "Linux", "media", "video", "JavaScript" and "Java", so the user is likely to be a Web developer or a fan of "media". We see that the recommended tags in Figure 12, such as "ajax", "videos" and "download", are highly related to the user's interests, which implies that the personalized browsing can indeed help users to find their interested resources effectively.

6.4.2 Time related browsing.

Figure 13 demonstrates the distribution of the tag counts associated with three URLs over time. It can be seen that given a certain URL, the popularity varies as time goes by. Our algorithm can discover the newly emerging resources such as URL 1 and URL 3 which increased to the peak at the beginning. It can also detect the resources which become the hot topics periodically like URL 2. At the same time, we found out that URL 3 was just created in Dec.21 2005 and became the hot topic in the following day, which means people can get the most popular topics by browsing the social annotation with time information.

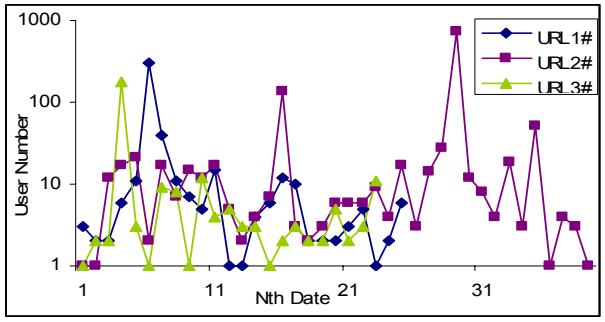

Figure 13. Illustration of Tag-URL coverage

Date 2005-12-12 http://www.nist.gov/dads/terms.html
Date 2006-2-28 http://developer.apple.com/tools/rubyonrails.html
Date2005-12-22http://www.exploding-boy.com/2005/12/21/more-free-css-navigation-menu-designs/

Table 2. Clustering Results

Programming	**Music**	**Science**	**Microsoft**
AJAX JavaScripts	Bittorrent Torrents	Health Sleep	XP Tweaks
Ruby rails	Ipod	Math Mathematics	Excel
PHP	Radio MP3	Physics Quantum	Word Writely
Python	Itunes	Psychology Brain	asp.net dotnet
Java	Guitar Chords	Space Astronomy	XBOX
Framework	Sound Soundfx	Algorithms	MSN WindowsLive
C Cpp	Player Songbird	MIT	Outlook
Dhtml	Indie	Biology	Boot Bootdisk
Lisp	Drm	Lectures sicp	Spyware
Perl	Lyrics song	Evolution Creationism	Vista Longhorn
Arts	**Basketball**	**Book**	**Computer**
Graffiti Streetart	ESPN	Lisp	Developers
Museum mus	Fox	Literature ebook	IE favorites
Knots topology	Autism	Audiobooks	Algorithms comupeterscience
Poetry Anvatagrade	Dallas	Amzaon	Spyware Adware
Artistis Painter	NBA	Scheme Sicp	
C	**Design**	**Google**	**Game**
Algorithm DataStructure	CSS Webdesign	Maps Googlemaps	Sudoku Puzzle
Cocoa Objective	Flash	Gmail	Warcraft worldofwarcraft
Mono	Art Graphics	GreasemonkeyUserscripts	Videogames
Compiler Compilers	Fonts Typography	Searchengines;	Chess
Visualstudio VS2005	Photoshop	GPS geocaching	Emulation emulators
Java	**Mac**	**Reference**	**Web**
Eclipse IDE	Ipod	CSS HTML	Css Webdesign
Framework	Itunes	Howto Tips	Ajax Javascript
Xmlhttprequest	Macosx OSX	Ebook ebooks	Google
J2EE Spring	Cocoa Objective	Maps	PHP
UML	Quicksilver	Language Dictionary	Del.icio.us Delicious

7. CONCLUSION AND FUTURE WORK

Social annotation browsing is a recently emerging task and becomes more and more important as the annotations of web resources keep on increasing at a surprising speed. In this paper, we analyze the characteristics of social annotations in three aspects, namely similarity, hierarchy and distribution. Based on observations in these aspects, we propose the ELSABer algorithm for effective social annotation browsing. A prototype system is also implemented based on ELSABer and produces encouraging results. Our main contributions can be concluded as follows:

- The proposal to study the problem of large scale annotation browsing. Previous methods can not work well when the annotation size is large.
- The proposal of the effective algorithm – ELSABer based on the analysis of social annotation's characteristics.
- The proposal of enhanced models for personalized and time related browsing.
- The implementation of a prototype system based on ELSABer to evaluate the proposed algorithm.

In the future, we will conduct more user studies for evaluating the effectiveness of our algorithm since browsing problem need more consideration in the view of user. Further more, we should emphasize on how to find more qualified URL resources and utilize existing hierarchical structures such as ODP and WordNet for helping construct more meaningful hierarchical structures for social annotations.

8. ACKNOWLEDGEMENT

The authors would like to thank IBM China Research Lab for its continuous support to and cooperation with Shanghai JiaoTong University. The authors also appreciate the valuable suggestions of Linyun Fu, Shengliang Xu and Lei Zhang. In the end, the authors would like to thank the three anonymous reviewers for their elaborate and helpful comments.

9. REFERENCES

[1] S. A. Golder, and B. A. Huberman. Usage Patterns of Collaborative Tagging Systems. *Journal of Information Science*, 32(2), pp.198-208, 2006

[2] J. Schachter. Del.icio.us about page. http://del.icio.us/doc/about, 2004

[3] T. Hammond, T. Hannay, B. Lund, and J. Scott. Social book marking tools (i) - a general review. *D-Lib Magazine*, 11(4), 2005.

[4] A. Mathes. Folksonomies – Cooperative Classification and Communication through Shared Metadata. http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html, December 2004

[5] E. Quintarelli. Folksonomies: power to the people. *Paper presented at the ISKO Italy-UniMIB meeting.* http://www.iskoi.org/doc/folksonomies.htm, June 2005

[6] X. Wu, L. Zhang, and Y. Yu. Exploring Social Annotations for the Semantic Web. In: *Proc. of WWW2006*, pp. 417-426, May 23.26, 2006

[7] C. H. Brooks, N. Montanez. Improved annotation of the blogosphere via autotagging and hierarchical clustering. In: *Proc. of WWW 2006*, pp. 625--632. May 23.26 2006

[8] P. A. Dmitriev, N. Eiron, M. Fontoura, and E. Shekita. Using Annotations in Enterprise Search. In: *Proc. of WWW2006* , pp. 811-817, May 23.26, 2006

[9] P. Mika Ontologies are us: a unified model of social networks and semantics. In: *Proc. of ISWC 2005*. pp. 522-536, Nov. 2005.

[10] M. Dubinko, R. Kumar, J. Magnani, J. Novak, P. Raghavan, A. Tomkins. Visualizing Tags over Time. In: *Proc. of WWW2006*, pp. 193-202, May 23.26, 2006

[11] C. Shirky. Folksonomy. Blog entry at http://www.corante.com/many/archives/2004/08/25/folksonomy.php, August 2004

[12] G. Smith. Atomiq: Folksonomy: social classification. http://atomiq.org/archives/2004/08/folksonomy_social_classification.html, Aug 3, 2004

[13] A. Pothen, H. D. Simon, and K.-P. Liou. Partitioning sparse matrices with eigenvectors of graphs. *In: SIAM J. Matrix Anal. Appl*., 11(3):430{452, 1990.

[14] J. B. MacQueen. Some Methods for classification and Analysis of Multivariate Observations, *In Proc.of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, 1:281-297 1967

[15] H. -J. Zeng, Q.-C. He, Z. Chen, W.-Y. Ma and J. Ma. Learning to cluster Web search results. In *Proc. of SIGIR 2004*, 2004

[16] http://www.freepatentsonline.com/5961593.html

[17] B. Christos, K. Vaggelis, M. Ioannis. A Web-page fragmentation technique for personalized browsing. In: *Proc. of the symposium on Applied computin*g pp. 1146 - 1147.

[18] http://www.stumbleupon.com/

[19] Q. Zhao, T.-Y. Liu, S.S. Bhowmick, and W.-Y. Ma. Event Detection from Evolution of Click-through Data. In *Proc. of SIGKDD 2006.*pp.484-493

[20] Del.icio.us tag cloud view: http://del.icio.us/tag/

[21] G. Begelman, P. Keller and F.Smadja, Automated Tag Clustering Improved search and exploration in the tag space In**:** *Proc. of Collaborative Web Tagging Workshop at WWW2006*.

[22] http://www.neuroticWeb.com/recursos/del.icio.us-graphs/

[23] http://hublog.hubmed.org/tags/visualisation