Personalization in Folksonomies Based on Tag Clustering

Jonathan Gemmell, Andriy Shepitsen, Bamshad Mobasher, Robin Burke

Center for Web Intelligence
School of Computing, DePaul University
Chicago, Illinois, USA

{jgemmell,ashepits,mobasher,rburke}@cti.depaul.edu

Abstract

Collaborative tagging systems, sometimes referred to as "folksonomies," enable Internet users to annotate or search for resources using custom labels instead of being restricted by pre-defined navigational or conceptual hierarchies. However, the flexibility of tagging brings with it certain costs. Because users are free to apply any tag to any resource, tagging systems contain large numbers of redundant, ambiguous, and idiosyncratic tags which can render resource discovery difficult. Data mining techniques such as clustering can be used to ameliorate this problem by reducing noise in the data and identifying trends. In particular, discovered patterns can be used to tailor the system's output to a user based on the user's tagging behavior. In this paper, we propose a method to personalize a user's experience within a folksonomy using clustering. A personalized view can overcome ambiguity and idiosyncratic tag assignment, presenting users with tags and resources that correspond more closely to their intent. Specifically, we examine unsupervised clustering methods for extracting commonalities between tags, and use the discovered clusters as intermediaries between a user's profile and resources in order to tailor the results of search to the user's interests. We validate this approach through extensive evaluation of proposed personalization algorithm and the underlying clustering techniques using data from a real collaborative tagging Web site.

Introduction

Collaborative tagging is an emerging trend allowing Internet users to manage and share online resources through user-defined annotations. There has been a recent proliferation of collaborative tagging systems. Two of the most popular examples are del.icio.us[1] and Flickr[2]. In del.icio.us users bookmark URLs. Flickr, on the other hand, allows users to upload, share and manage pictures. Other applications specialize in music, blogs, or journal publications.

At the foundation of collaborative tagging is the annotation; a user describes a resource with a tag. A collection of annotations results in a complex network of interrelated users, resources and tags, commonly referred to as a folksonomy (Mathes 2004). Users are free to navigate through the folksonomy without being tied to a pre-defined navigational or conceptual hierarchy. The freedom to explore this large information space of resources, tags, or even other users is central to the utility and popularity of collaborative tagging. Tags make it easy and intuitive to retrieve previously viewed resources (Hammond et al. 2005). Further, tagging allows users to categorized resources by several terms, rather than one directory or a single branch of an ontology (Millen, Feinberg, and Kerr 2006). Collaborative tagging systems have a low entry cost when compared to systems that require users to conform to a rigid hierarchy. Furthermore, users may enjoy the social aspects of collaborative tagging (Choy and Lui 2006). Collaborative tagging offers a sense of community, not provided by either ontologies or search engines. Users may share or discover resources through the collaborative network and connect to people with similar interests.

Because collaborative tagging applications reap the insights of many users rather than a few "experts", they are more dynamic and able to incorporate a changing vocabulary or absorb new trends quickly (Wu, Zhang, and Yu 2006). These applications can identify groups of like-minded users, catering not only to mainstream but also to non-conventional users that are often under-served by traditional Web tools. Search engines, the most widely used tool for searching large information spaces, attempts to index resources. In effect, this is a pull-model, where the application pulls resources from the information space (Yan, Natsev, and Campbell 2007). In contrast, collaborative tagging applications support a push-model: users identify which resources are relevant and through the annotation process promote the resource. Consequently, the collaborative tagging system may be populated with resources a pull-model may not be able to locate.

A collaborative tagging application may be applied to a variety of resource: Web pages, news stories, pictures, video clips, etc. The content of some of these resources can be determined by computers, but some resources are particularly difficult to automatically categorize except by relying on meta-data. For example, it can be particularly difficult for standard search engines to describe or organize video and music resources. Collaborative Tagging applications, however, rely on the user's tags to determine the content of a resource. As such, it may be easier for users to locate

[1] del.icio.us

[2] www.flickr.com

resources with collaborative tagging applications than with standard search engines.

Another advantage of collaborative tagging applications is the richness of the user profiles. As users annotate resources, the system is able to track their interests. Data mining tools such as clustering can identify important trends and characteristics of the users. These profiles are a powerful tool for personalization algorithms (Yan, Natsev, and Campbell 2007). The relatively low cost of generating a user profile (typically based on the resources and tags associated with a user), can be dramatically overcome by the improved user experience personalization allows. The benefit of personalization in search to the user is described in (Teevan, Dumais, and Horvitz 2007).

Even though collaborative tagging applications have many benefits, they also present unique challenges for search and navigation. Most collaborative tagging applications permit unsupervised tagging; users are free to use any tag they wish to describe a resource. This is often done to reduce the entry cost of using the application and make collaborative tagging more user friendly. As a result, folksonomies contain a wide variety of tags: from the factual (e.g., "Mt Rushmore") to the subjective (e.g., "boring"), and from the semantically-obvious (e.g., "Chicago") to the utterly opaque (e.g., "jfgwh"). Moreover, tag redundancy in which several tags have the same meaning or tag ambiguity in which a single tag has many different meanings can confound users searching for resources. The task of combating noise is made even more difficult by capitalization, punctuation, misspelling, and other discrepancies.

Data mining techniques such as clustering provide a means to overcome these problems. Through clustering, redundant tags can be aggregated; the combined trend of a cluster can be more easily detected than the effect of a single tag. The effect of ambiguity can also be diminished, since the uncertainty of a single tag in a cluster can be overwhelmed by the additive effects of the rest of the tags. Tag clusters may represent coherent topic areas. By associating a user's interest to a particular cluster, we may surmise the user's interest in the topic.

Personalization can also be used to overcome noise in folksonomies. Given a particular user profile, the user's interests can be clarified and navigation within the folksonomy can be tailored to suit the user's preferences. By using both clustering and personalization together we seek to combat noise in folksonomies and improve the user experience.

In this paper, we propose an algorithm to personalize search and navigation based on tags in folksonomies. The core of our algorithm is a set of tag clusters, discovered based on their associations with resources by various users. The personalization algorithm models users as vectors over the set of tags. By measuring the importance of a tag cluster to a user, the user's interests can be better understood and the context of user's interaction can be better delineated. Likewise, each resource is also modeled as a vector over the set of tags. By associating resources with tag clusters, resources relevant to the topics captured by those clusters can be identified. By using the tag clusters as intermediaries between a user and a resource, we infer the relevance of the resource to the user. We then use the inferred relevance of the resource to re-rank the results of a basic search, thereby personalizing the user experience.

Furthermore, we evaluate the effectiveness of several clustering techniques that can be used as part of the personalized search and navigation framework: hierarchical agglomerative clustering, maximal complete link clustering, and k-means clustering. Hierarchical clusters, in particular, offers more flexibility than other clustering methods. By selecting clusters in the hierarchy directly related to the user's action, the algorithm may focus more clearly on the user's intent. Alternatively, by including more clusters, the result can be generalized promoting serendipitous discovery.

The rest of this paper is organized as follows. We begin with presenting some related work involving the use of clustering and personalization in folksonomies. We then outline basic approaches used for search in folksonomies and motivate the need for personalization. The clustering methods and our personalization algorithm are then presented. In the experimental evaluation section we evaluate our personalization algorithm and compare the effectiveness of clustering algorithms in that context. Finally, we conclude the paper and offer some directions for future work.

Related Work

A fundamental assumption in this paper is the ability of clustering algorithms to form coherent clusters of related tags. Support for that assumption is given in (Begelman, Keller, and Smadja 2006) where tag clustering is suggested to improve search in folksonomies and (Heymann and Garcia-Molina April 2006) where hierarchical agglomerative clustering is proposed to generate a taxonomy from a folksonomy.

Integral to our algorithm for personalization using clusters is the measurement of relevance between a user and a resource. A similar notion was previously described in (Niwa, Doi, and Honiden 2006) in which an affinity level was calculated between a user and a set of tag clusters. A collection of resources was then identified for each cluster based on tag usage. Resources were recommended to the user based on the user's affinity to the clusters and the associated resources.

Our algorithm relies heavily on tag clusters and the utility they offer, but clusters have many other potential functions worth noting. Tag clusters could serve as intermediaries between two users in order to identify like-minded individuals allowing the construction of a social network. Tag clustering can support tag recommendation, reducing annotation to a mouse click rather than a text entry. Well chosen tags make the recovery process simple and offer some control over the tag-space. By exerting some control over the tag space, the effect of tag redundancy or ambiguity can be mitigated to some degree. In (Xu et al. 2006) a group of tags are offered to the user based on several criteria (coverage, popularity, effort, uniformity) resulting in a cluster of a relevant tags.

Clustering is an important step in many attempts to improve search and navigation. In (Wu, Zhang, and Yu 2006), tag clusters are presumed to be representative of the resource content. Thus, a folksonomy of Web resources is used to

move the Internet closer to the Semantic Web. In (Choy and Lui 2006) a two-dimensional tag map is constructed. In this manner, tag clusters can be used as waypoints in the tag space and facilitate navigation through the folksonomy. In (Hayes and Avesani 2007), topic relevant partitions are generated by clustering resources rather than tags. Then the most characteristic resources of the clusters are identified. Users interested in the topic represented by a cluster may be particularly interested in the characteristic resources.

By using clusters of resources, Flickr, a popular collaborative tagging application for pictures, improves search and navigation by discriminating between different meanings of a user query. For example, a user selecting the tag "apple" will receive several groups of pictures. One group represents "fruit"; while another contains iPods, iMacs, and iPhones. A third cluster contains pictures of New York City. In (Chen and Dumais 2000) clusters of resources are shown to benefit navigation by categorizing the resources into topic areas. An advantage of this approach is that the user may interactively disambiguate his query.

Search and Navigation in Folksonomies

In traditional Internet applications the search and navigation process serves two vital functions: retrieval and discovery. Retrieval incorporates the notion of navigating to a particular resource or a resource containing particular content. Discovery, on the other hand, incorporates the notion of finding resources or content interesting but theretofore unknown to the user. The success of collaborative tagging is due in part to its ability to facilitate both these functions within a single user-centric environment.

Reclaiming previously annotated resources is both simple and intuitive, as most collaborative tagging applications often present the user's tag in the interface. Selecting a tag displays all resources annotated by the user with that tag. Users searching for particular resources they have yet to annotate may select a relevant tag and browse resources annotated by other users. However, the discovery process can be much more complex. A user may browse the folksonomy, navigating through tags, resources, or even other users. Furthermore, the user may select one of the results of a query (i.e. tag, resource, or user) as the next query itself. This ability to navigate through the folksonomy is one reason for the popularity of collaborative tagging. While the user can navigating through many different dimensions of a folksonomy, this work focuses on searching for resources using a tag as query.

A folksonomy can be described as a four-tuple D:

$$D = \langle U, R, T, A \rangle, \tag{1}$$

where, U is a set of users; R is a set of resources; T is a set of tags; and A is a set of annotations, represented as user-tag-resource triples:

$$A \subseteq \{\langle u, r, t \rangle : u \in U, r \in R, t \in T\} \tag{2}$$

A folksonomy can, therefore, be viewed as a tripartite hyper-graph (Mika 2007) with users, tags, and resources represented as nodes and the annotations represented as hyper-edges connecting a user, a tag and a resource.

Standard Search in Folksonomies

Contrary to traditional Internet applications, a search in a collaborative tagging application is performed with a tag rather than a keyword. Most often the tag is selected through the user interface.

Applications vary in the way they handle navigation. Possible methods include recency, authority, linkage, or vector space models (Salton, Wong, and Yang 1975). In this work we focus on the vector space model adapted from the information retrieval discipline to work with folksonomies. Each user, u, is modeled as a vector over the set of tags, where each weight, $w(t_i)$, in each dimension corresponds to the importance of a particular tag, t_i.

$$\vec{u} = \langle w(t_1), w(t_2) ... w(t_{|T|}) \rangle \tag{3}$$

Resources can also be modeled as a vector over the set of tags. In calculating the vector weights, a variety of measures can be used. The *tag frequency*, *tf*, for a tag, *t*, and a resource, *r* is the number of times the resource has been annotated with the query tag. We define *tf* as:

$$tf(t,r) = |\{a = \langle u, r, t \rangle \in A : u \in U\}| \tag{4}$$

Likewise, the well known *term frequency * inverse document frequency* (Salton and Buckley 1988) can be modified for folksonomies. The *tf*idf* multiplies the aforementioned frequency by the relative distinctiveness of the tag. The distinctiveness is measured by the log of the total number of resources, *N*, divided by the number of resources to which the query tag was applied, n_t. We define *tf*idf* as:

$$tf*idf(t,r) = tf(t,r) * \log(N/n_t) \tag{5}$$

With either term weighting approach, a similarity measure between a query, *q*, represented as a vector over the set tags, and a resource, *r*, also modeled as a vector over the set tags, can be calculated. However, in this work, since search or navigation is often initiated by selecting a single tag from the user interface, we assume the query is a vector with only one tag.

Several techniques exist to calculate the similarity between vectors such as Jaccard similarity coefficient or Cosine similarity (Van Rijsbergen 1979). Cosine similarity is a popular measure defined as:

$$cos(q,r) = \frac{\sum_{t \in T} tf(t,q) * tf(t,r)}{\sqrt{\sum_{t \in T} tf(t,q)^2} * \sqrt{\sum_{t \in T} tf(t,r)^2}} \tag{6}$$

A basic search may begin by calculating the cosine similarity of the query to each resource. Once the similarity is calculated, an ordered list can be returned to the user. Since the query is modeled as a vector containing only one tag, this equation may be further simplified. However, we have provided the full equation so that it can be applicable in a more general setting. Other characteristics of the resource such as recency or popularity can be used to augment the query. In this work we focus on cosine similarity.

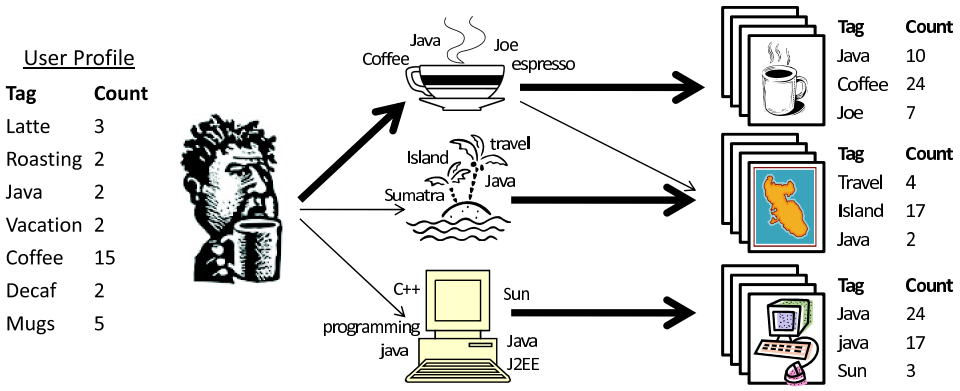

Figure 1: Clusters represent coherent topic areas and serve as intermediaries between a user and the resources.

Need for Personalization

A standard search does not take into account the user profile and returns identical results regardless of the user. While personalization has been shown to increase the utility of Web applications, the need for personalization in folksonomies is even more critical. Noise in the folksonomy, such as tag redundancy and tag ambiguity, obfuscate patterns and reduce the effectiveness of data mining techniques. Redundancy occurs when two users apply different tags with identical meaning (e.g. "java" and "Java"). Redundant tags can hinder algorithms that depend on calculating similarity between resources. A user searching with a redundant tag may not find resources annotated with another similar tag. Ambiguity occurs when two users apply an identical tag, but mean something different (e.g. "java" applied to a www.starbucks.com and "java" applied to www.sun.com). Ambiguous tags can result in the overestimation of the similarity of resources that are in fact unrelated. A user searching with an ambiguous tag may receive results unrelated to his intended meaning.

Tag clustering provides a means to combat noise in the data and facilitate personalization. By aggregating tags into clusters with similar meaning, tag redundancy can be assuaged since the trend for a cluster can be more easily identified than the effect of a single tag. Ambiguity will also be remedied to some degree, since a cluster of tags will assume the aggregate meaning and overshadow any ambiguous meaning a single tag may have. Furthermore, using clusters to represent a topic area, the user's interest in that topic can be more easily quantified. The connection of a resource to a tag cluster can also be quantified. If tag clusters are used as a nexus between users and resources, the users interest in resources can be calculated. Consequently results from a basic search can be re-ranked to reflect the user profile.

Personalization, therefore, is also critical in our attempts to combat noise in the data. The user's intended meaning for ambiguous tags can be inferred through the analysis of other tags and resources in the user profile. Therefore, even though a user may annotate resources with redundant tags, personalization techniques may reduce the noise generated by these tags.

Personalized Search Based on Tag Clustering

In this section, our algorithm for search personalization based on tag clustering is described in detail. We first offer a brief overview of the proposed approach. Then we describe several methods for clustering and their parameters. We next describe the personalization algorithm and how the discovered clusters are used to connect a user to a resource, providing a means to measure the relevance of a resource to a user. Finally, we show how the results of a basic search strategy can be personalized by incorporating the relevance of the resources to the user.

Overview of the Proposed Approach

In our approach, tag clusters serve as intermediaries between a user and the resources. Once a set of tag clusters are generated, the user's interest in each cluster is calculated. A strong interest indicates the user has frequently used the tags in the cluster. Likewise, a measure is calculated from each cluster to all resources. A strong relationship between a tag cluster and a resource means many of the tags were used to describe the resource.

By using clusters to connect the user to the resources, the relevance of the resource to the user can be inferred. Once a relevance measure is calculated for all resources, a list of resources provided by a traditional search can be reordered and presented to the user. Each user, therefore, receives a personalized view of the information space.

For example, in Figure 1, a hypothetical user searching based on the tag "Java" has a strong connection to the cluster of coffee related tags, and weaker connections to the clusters dealing with traveling or computer programming. The strength of the connection is based on the similarity of the user profile to the tag clusters. Likewise, the resources dealing with coffee has a strong relation to the coffee cluster. The user's interest in that resource can, therefore, be inferred. However, the coffee cluster has a weak relation to resources dealing with travel to Java and Sumatra. The rel-

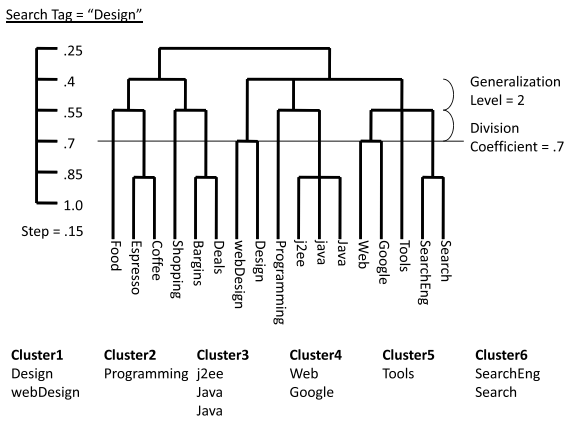

Figure 2: An example of hierarchical tag clustering.

evance of the traveling resources to the user is consequently minimal. Because of the crucial role tag clusters play in the personalization algorithm, we evaluate several clustering techniques and their parameters in order to achieve the maximum results.

Tag Clustering

A critical element of our algorithm is a set of tag clusters that connects a user with the resources. For many clustering techniques, the similarity between tags must first be calculated. The cosine similarity between two tags, t and s, may be calculated by treating each tag as a vector over the set of resources and using *tag frequency* or *tag frequency * inverse document frequency* as the weights in the vectors.

$$cos(t,s) = \frac{\sum_{r \in R} tf(t,r) * tf(s,r)}{\sqrt{\sum_{r \in R} tf(t,r)^2} * \sqrt{\sum_{r \in R} tf(s,r)^2}} \quad (7)$$

Once the cosine similarities are calculated, it is possible to construct clusters. Our personalization method is independent of the clustering technique. In this paper we evaluate three clustering techniques: a specific version of hierarchical agglomerative clustering, maximal complete link clustering and k-means clustering.

Hierarchical Agglomerative Clustering One of the algorithms we consider is a version of hierarchical agglomerative clustering (Gower and Ross 1969) modified to suit out needs for tag clustering. As the hierarchical clustering algorithm begins each tag forms a singleton cluster. Then, during each stage of the procedure, clusters of tags are joined together depending on the level of similarity between the clusters. This is done for many iterations until all tags have been aggregated into one cluster. The result is a hierarchical clustering of the tags such as the one depicted in Figure 2.

Several techniques exist to calculate the similarity between tag clusters and to merge smaller clusters. The minimum distance between any tag from one cluster to any tag from another cluster could be used (often called single link). Likewise, the maximum distance could be used (often called complete link). It is also possible to calculate the similarity between every tag in one cluster and every tag in the other cluster and then take the average of these similarities. For this work, we focus on using the latter centroid-based approach which is not as computationally expensive as the other techniques.

To compute the similarity between clusters, a centroid for each cluster is calculated. Each tag is treated as a vector over the set resources. Vector weights are calculated using either *tf* or *tf*idf*. The centroid for a cluster is calculated as the average vector of the tag's vectors. The similarity between two clusters is then calculated using the centroids as though they were single tags.

Hierarchical agglomerative clustering has several parameters that require tuning in order to achieve optimum results in the personalization routine. The parameter *step* is the decrement by which the similarity threshold is lowered. At each iteration, clusters of tags are aggregated if the similarity between them meets a minimum threshold. This threshold is lowered by *step* at each iteration until it reaches 0. By modifying this parameter the granularity of the hierarchy can be controlled.

In order to break the hierarchy into distinct clusters, a *division coefficient* is chosen as a cutoff point. Any cluster below this similarity threshold is considered an independent cluster in the personalization routine. Selecting a value near one will result in many small clusters with high internal similarity or possibly even singletons. Alternatively, selecting a small value will result in fewer larger clusters, with lower internal similarity.

An important modification to the traditional hierarchical clustering method is the *generalization level*. Normally, all clusters below the *division coefficient* would be used, but in this modification only those clusters descendent of the selected tag are used. The *generalization level* allows the algorithm to return more general tag clusters for the hierarchy. Instead of using only the descendants of the selected tag, a larger branch of the hierarchy is used by first traveling up the tree the specified number of levels. Notice that if the *generalization level* is set very high it will include all clusters in the hierarchy and behave as a traditional agglomerative clustering algorithm.

For example, in the hypothetical hierarchy of clusters depicted in Figure 2, if the user selects the tag "Design," the algorithm will first identify the level at which this tag was added to the hierarchy. In this case, it was added when the similarity threshold was lowered to .7. With a *generalization coefficient* of 2, the algorithm proceeds up two levels in the hierarchy. Finally, the *division coefficient* is used to break the branch of the hierarchy into distinct clusters.

In order to ascertain the relative value of the modified hierarchical clustering technique, two other clustering methods were evaluated: maximal complete link clustering and k-means clustering.

Maximal Complete Link Clustering Maximal complete link clustering identifies every maximal clique in a graph (Augutson and Minker 1970). A maximal clique is a clique

that it not contained in a larger clique. Maximal complete link clustering permits clusters to overlap. This may be particularly advantageous when dealing with ambiguous tags. The tag "java" for example could be a member of a coffee cluster as well as programming cluster.

In this work maximal complete link clusters were constructed using a branch and bound method. Maximal complete link clustering is a well known NP-hard problem. Fortunately, the extreme sparsity of the data permits the application of this method, since the number of potential solution is dramatically reduced. Nevertheless, this method was the most time intensive of the clustering methods we evaluated. Fortunately, clustering in the proposed personalization algorithm clustering is done off-line. But, it will not scale well to larger datasets. Approximation techniques could be used to save computational time at the expense of missing some clusters (Johnson 1973).

Maximal complete link clustering has one parameter to tune, the *minimum similarity threshold*. If the similarity between two tags meets this threshold, they are considered to be connected. Otherwise, they are considered to be disconnected. Treating each tag as a node, a sparse graph is generated. From this graph the maximal complete clusters are discovered.

k-means Clustering The last clustering approach used to evaluate the usefulness of clustering to the personalization algorithms was k-means clustering (MacQueen 1967). A predetermined number of clusters, *k*, are randomly populated with tags. Centroids are calculated for each cluster. Then, each tag is reassigned to a cluster based on a similarity measure between itself and the cluster centroid. Several iterations are completed until tags are no longer reassigned. This clustering method has only one parameter to tune, k. The relatively efficient computational time of the algorithm makes k-means an attractive alternative.

In contrast to hierarchical or maximal complete link clustering, k-means clustering cannot effectively isolate irrelevant tags. In maximal complete link clustering, tags with a very weak connection (or no connection) can be isolated in a singleton cluster. Such a cluster has little effect on the personalization algorithm. Similarly, in hierarchical agglomerative clustering, tags with a strong connection are identified first. Weakly connected tags are not aggregated until the last stages of the algorithm. Further, by modifying the division coefficient, these weak connections and the irrelevant tags can be entirely ignored. The k-means algorithm, however, includes all tags in one of the k clusters and each tag in a cluster is given equal weight. If a cluster contains tags covering several topic areas the aggregate meaning of the cluster can become muddled. Or, if a cluster contains tags irrelevant to the consensus meaning, the aggregate meaning in the cluster can become diminished. This drawback may overshadow the benefit of faster computational time.

The clustering method is independent from the personalization algorithm; any clustering approach could be used. Clusters are, however, integral to the algorithm. We next show how the personalization algorithm uses clusters to bridge the gap between users and resources.

Personalization Algorithm Based on Tag Clustering

There are three inputs to a personalized search: the selected tag, the user profile and the discovered clusters. The output of the algorithm is an ordered set of resources.

For each cluster, c, the user's interest is calculated as the ratio of times the user, u, annotated a resource with a tag from that cluster over the total number of annotations by that user. We denote this weight as *uc_w(u,c)* and defined it as:

$$\textit{uc\_w(u,c)} = \frac{|\{a = \langle u, r, t\rangle \in A : r \in R, t \in c\}|}{|\{a = \langle u, r, t\rangle \in A : r \in R, t \in T\}|} \quad (8)$$

Also, the relation of a resource, r, to a cluster is calculated as the ratio of times the resource was annotated with a tag from the cluster over the total number of times the resource was annotated. We call this weight *rc_w(r,c)* and defined it as:

$$\textit{rc\_w(r,c)} = \frac{|\{a = \langle u, r, t\rangle \in A : u \in U, t \in c\}|}{|\{a = \langle u, r, t\rangle \in A : u \in U, t \in T\}|} \quad (9)$$

Both *uc_w(u,c)* and *rc_w(r,c)* will always be a number between zero and one. A higher value will represent a strong relation to the cluster.

The relevance of the resource to the user, *relevance(u,r)*, is calculated from the sum of the product of these weights over the set of all clusters, C. This measure is defined as:

$$\textit{relevance(u,r)} = \sum_{c \in C} \textit{uc\_w(u,c)} * \textit{rc\_w(r,c)} \quad (10)$$

Intuitively, each cluster can be viewed as the representation of a topic area. If a user's interests parallels closely the subject matter of a resource, the value for *relevance(u,r)* will be correspondingly high as can be seen in Figure 1.

To improve performance at the expense of memory the relevance of each resource to every user may be precalculated. In fact, at this point the relevance matrix may be useful in its own right, perhaps for personalizing recommendations or comparing the interests of two users. In order to personalize search and navigation, the selected tag is taken into account.

A basic search is performed on the query, q, using the vector space model and *tag frequency*. A similarity, *rankscore(q,r)*, is calculated for every resource in the dataset using the cosine similarity measure. A personalized similarity is calculated for each resource by multiplying the cosine similarity by the relevance of the resource to the user. We denote this similarity as *p_rankscore(u,q,r)* and define it as:

$$\textit{p\_rankscore(u,q,r)} = \textit{rankscore(q,r)} * \textit{relevance(u,r)} \quad (11)$$

Once the *p_rankscore(u,q,r)*, has been calculated for each resource, the resources are returned to the user in descending order of the score. While the weights from clusters to resources will be constant regardless of the user, the weights connecting the users to the clusters will differ based on the user profile. Consequently, the resulting *p_rankscore(u,q,r)* will depend on the user and the results will be personalized.

Experimental Evaluation

We validate our approach through extensive evaluation of the proposed algorithm using data from a real collaborative tagging Web site. A Web crawler was used to extract data from del.icio.us from 5/26/2007 to 06/15/2007. In this collaborative tagging application, the resources are Web pages. The dataset contains 29,918 users, 6,403,442 resources and 1,035,177 tags. There are 47,184,492 annotations with one user, resource and tag. A subset of the users in the dataset was used as test cases. The annotations of the remaining users, were used to generate the tag clusters.

Each test case consisted of a user, a tag and a resource. After performing a basic search using only the tag, the rank of the resource in the returned results was recorded. Next, a personalized search using the same tag was performed, taking into account the user profile and the discovered clusters. The rank of the resource in the personalized search results was also recorded.

Since the resource was annotated by the user, we assume it is indeed of interest to the user. By comparing the rank of the resource in the basic search to the rank in the personalized search, we measured the effectiveness of the personalization algorithm. This section describes, in more detail, the process for judging the proposed personalization algorithm, and supplies an evaluation of the results.

Examples of Tag Clusters

The clustering algorithm is independent of the personalization algorithm. Still, the quality of the clusters is crucial to the success of the personalization algorithm. We assume that tags can be clustered into coherent clusters representing distinct topic areas. Support for that assumption is given in Table 1 where representative tags (those closest to the cluster centroids) were selected from six of our discovered tag clusters.

Cluster 1 represents the notion of literature and citations, while cluster 4 represents the notion of testing an Internet connection. Other clusters show clearly recognizable categories. Clusters can capture misspellings, alternative spellings or multilingualism such as in cluster 2: "paleta" versus "palette." They can also capture other redundant tags that are not variations of a particular word such as in cluster 5: "concerts" and "band."

Cluster 4 shows how users may have annotated resources with possibly either "speed testing" or "speedtest." Often collaborative tagging applications will treat the former as two individual tags. Still, the clustering algorithm will capture the similarity. Users interested in "speedtest" are likely to be also interested in resources annotated with both "speed" and "test." Most ontologies prohibit such ambiguity from entering the system by enforcing a set of rules. Folksonomies, however, because of their open nature are prone to this type of noise. Clustering appears to be a viable means to alleviate some of this redundancy.

Cluster 3, on the other hand, shows how clustering can generate odd relationships. The tags "peace" and "war" have been clustered together since the two words are often used together in the same context. Yet, they have entirely different meanings. Further, it is interesting to note the inclusion of "orwell" and "1984" with "fascism." The former has a strong literature context, while the latter has a social-political context. Clustering has nevertheless identified the similarities across these two contexts. Serendipity can be promoted by identifying such relations the user might otherwise be unaware of.

Cluster 1	**Cluster 2**	**Cluster 3**
linguistics	graphical	peace
literacy	linea	war
papers	paleta	1984
anthropology	palette	terrorism
bibliographies	rgb	fascism
thesis	hexadecimal	orwell
Cluster 4	**Cluster 5**	**Cluster 6**
speed	live	financial
testing	concerts	retirement
test	band	savings
broadband	group	finances
bandwidth	event	wealth
speedtest	party	plan

Table 1: Examples of tag clusters

Another assumption of the proposed approach is the ability to correctly identify resources with tag clusters. Support for that assumption is given in Table 2. Six strongly related Web pages were selected for each clusters from Table 1 (based on their relative weights in the centroid vectors for each cluster).

Web pages for Cluster 2 all share the notion of color and design and relate well to the tag cluster. For example, users having annotated "colorblender.com" are also likely to be interested in "kuler.adobe.com." Both retrieval and discovery are well served by these clusters. It is obvious that a user selecting "terrorism" from the user interface might wish to view "globalincidentmap.com." Less obvious is the relevance of "pickthebrain.com/blog." However, the strong relation of this resource to cluster 3 can support serendipitous discovery and improve the user's navigational experience.

Experimental Methodology

With our assumptions for the utility of tag clustering verified we turn our attention to the personalization algorithm. Figure 3 shows the experimental procedure. Two random samples of 5,000 users were taken from the dataset. Five-fold cross validation was performed on each sample. For each fold, 20% of the users were partitioned from the rest as test users. Clustering was completed using the data from the remaining 80% of the users. Clusters were generated using hierarchical, maximal complete link and k-means clustering. Optimum values for the relevant parameters of each method were derived empirically.

From each user in the test set, 10% of the user's annotations were randomly selected as test cases. Each case consisted of a user, tag and resource.

The basic search requires only a tag as an input. Re-

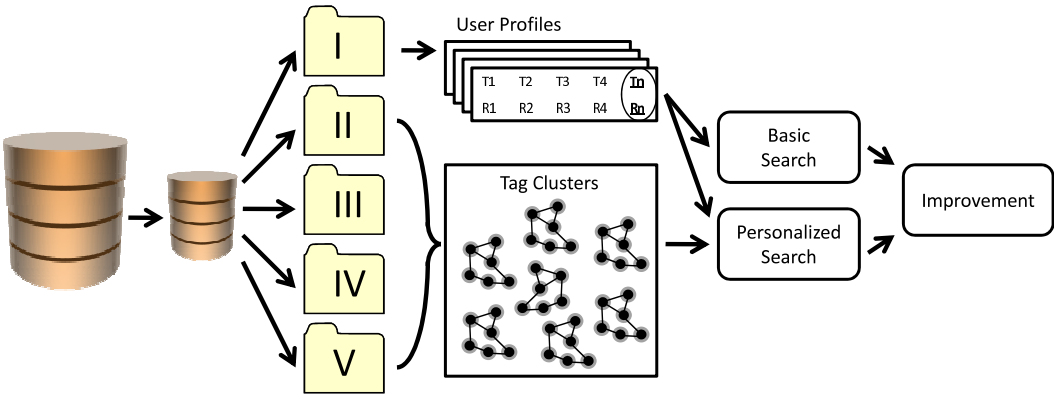

Figure 3: The steps used to test the personalization algorithm.

Cluster 1	**Cluster 2**
citeulike.org	colorblender.com
accent.gmu.edu	kuler.adobe.com
swoogle.umbc.edu	picnik.com
citeseer.ist.psu.edu	visuwords.com
www.eee.bham.ac.uk	pingmag.jp
oedb.org	touchgraph.com
Cluster 3	**Cluster 4**
studentsfororwell.org	speakeasy.net/speedtest
kirjasto.sci.fi/gorwell.htm	speedtest.net
adbusters.org/media/flash	speedtest.net/index.php
pickthebrain.com/blog	bandwidthplace.com
globalincidentmap.com	gigaom.com
secularhumanism.org	bt.com
Cluster 5	**Cluster 6**
eventful.com	thesimpledollar.com
pageflakes.com	money.aol.com/savings
gethuman.com/us	aaronsw.com
alamiracom.multiply.com	annualcreditreport.com
wamimusic.com/events	globalrichlist.com
torrentreactor.net	finance.yahoo.com

Table 2: Examples of resources strongly related to the tag clusters

sources were modeled as vectors over the set of tags. Similarly, the test tag was treated as a vector containing only one tag. The cosine similarity was calculated for all resources to the test tag, and the resources were then ordered. The rank of the resource in the basic search, r_b, was recorded.

The personalized search requires the test tag, the test user profile, and a set of discovered clusters. The relevance of the resources to the test user was calculated and was used to re-rank the resources. The new rank of the test resource in the personalized search, r_p, was recorded. Since the user has annotated this resource, we assume that it is relevant to the user in the context of the test tag. A personalized search should improve the ranking of the resource.

In order to judge the improvement provided by the personalized search, the difference in the inverse of the two ranks can be used, imp (Voorhees 1999). It is defined as:

$$imp = \frac{1}{r_p} - \frac{1}{r_b} \tag{12}$$

If the personalized approach re-ranks the resource nearer the top of the list, the improvement will be positive. Similarly, if the personalized approach re-ranks the resource further down the list, the improvement will be negative. In one extreme, if the basic search ranks the resource very low and the personalization algorithm improves its rank to the first position, then imp will approach one. The value for imp can never be greater than one. In practice, however, it is difficult for a personalization algorithm to achieve this level of success, since the potential improvement is bounded by the rank of the basic search. If a basic search ranks the resource in the forth position, the best improvement a personalized search can achieve is .75.

For each clustering technique and parameter for the technique, the improvement across all folds and samples was averaged and reported in the results below.

Experimental Results

In general, the proposed personalization technique results in improved performance, ranking resources known to be relevant to the user nearer the top of the search results. While all clustering techniques showed improvements, they did so with varying degrees. In addition, the input parameters were shown to have a marked effect on the final results.

The choice of tf or $tf * idf$ also played an important role. In all cases $tf * idf$ is superior. This is likely the result of the normalization that occurs. For example, the effect of a tag with little descriptive power such as “cool” or “toBuy” is diminished when compared to other tags with strong descriptive power such as “concerts” or “retirement.” The two weighting techniques appear to have nearly identical trends in the tuning of the parameters.

Hierarchical agglomerative clustering produced superior performance when compared to the other clustering methods, perhaps due to its inherent flexibility. The input parameters, *step*, *division coefficient* and *generalization level*, offer

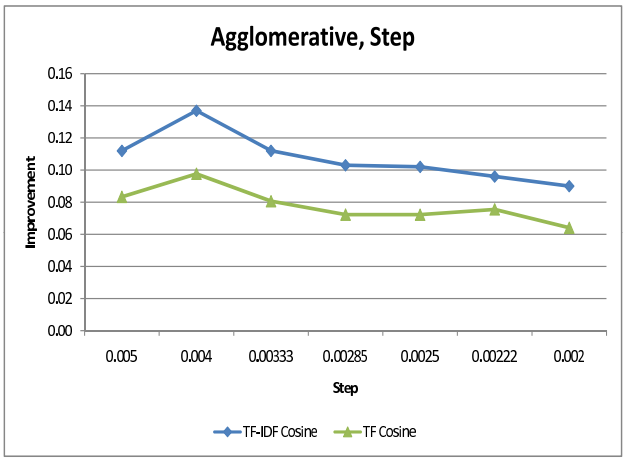

Figure 4: The effect of *step* in hierarchical clustering on the personalization algorithm.

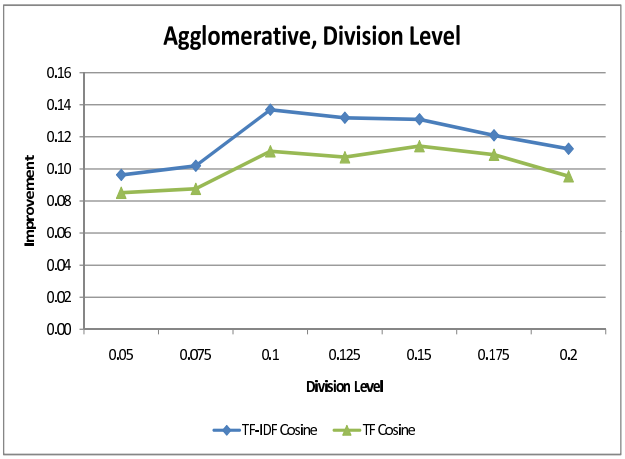

Figure 5: The effect of *division level* in hierarchical clustering on the personalization algorithm.

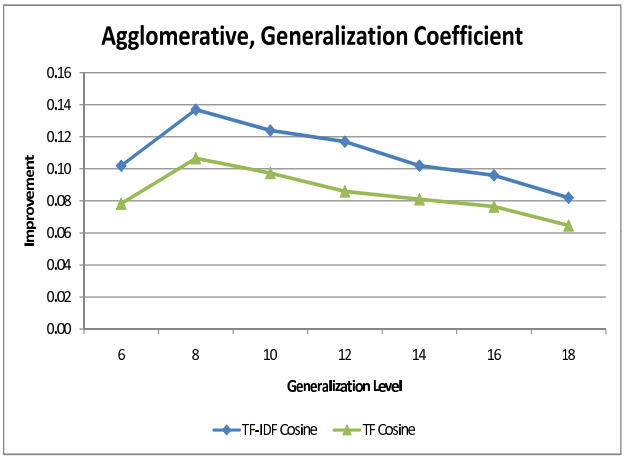

Figure 6: The effect of *generalization level* in hierarchical clustering on the personalization algorithm.

a level of tuning that the other two clustering methods could not provide.

The parameter, *step*, controls the granularity of the derived agglomerative clusters. The similarity threshold, initially set to one, is reduced by the value *step* at each iteration until it reaches zero. Clusters of tags are aggregated together if their similarity measure meets the current threshold. An ideal value for *step* would aggregate tags slowly enough to capture the conceptual hierarchy between individual clusters. For example, "Java" and "J2EE" should be aggregated together before they are in turn aggregated with "programming."

In Figure 4, increasing the value of *step* results in diminished performance, as tags are aggregated too quickly. Yet, if the value for step is too low, the granularity of the derived clusters can become too fine grained and overspecialization can occur. In these experiments, best results were achieved with a value of 0.004 as is shown in Figure 4. The same value for *step* was used when testing the other parameters.

The *division coefficient* plays a crucial role in the agglomerative clustering routine. It defines the level where the hierarchy is dissected into individual clusters. In the example provided in figure 2, the division coefficient is 0.7. Clusters below this level in the hierarchy are considered independent.

If the *division coefficient* is set too low, the result is a few large clusters. These clusters may include many relatively unrelated tags and span several topic areas. Likewise, if the *division coefficient* is set too high, the result will be many small clusters. While the tags in these clusters may be very similar, they might not aggregate together all the tags necessary to describe a coherent topic in the way a larger cluster could.

Since, the personalization algorithm relies on these clusters to serve as the intermediary between users and resources and presupposes the clusters represent distinct well defined topics, the selection of the *division coefficient* is integral to the success of the personalization algorithm. Intuitively, the goal of tuning the *division coefficient* is to discover the optimum level of specificity. As shown in Figure 5, the optimum value for this dataset is approximately 0.1. This is also the value used when testing other parameters.

The *generalization level* is key to our modification of the hierarchical agglomerative clustering algorithm. Normally, every cluster below the *division coefficient* would be returned by the clustering algorithm. In our modification, however, we select clusters directly related to the user's action. First, the position in which the test tag was aggregated into the hierarchy is noted. Then the algorithm returns only those cluster descendent of the test tag. However, we may include a broader swath of clusters by first traveling up the hierarchy by the *generalization level* and cutting off a larger branch as is shown in Figure 2.

The importance of the *generalization level* is demonstrated in Figure 6. If the *generalization level* is set too low, it is possible to overlook a cluster representing relevant resources. However, if the algorithm includes too many clusters not related to the user's selected tag, irrelevant factors

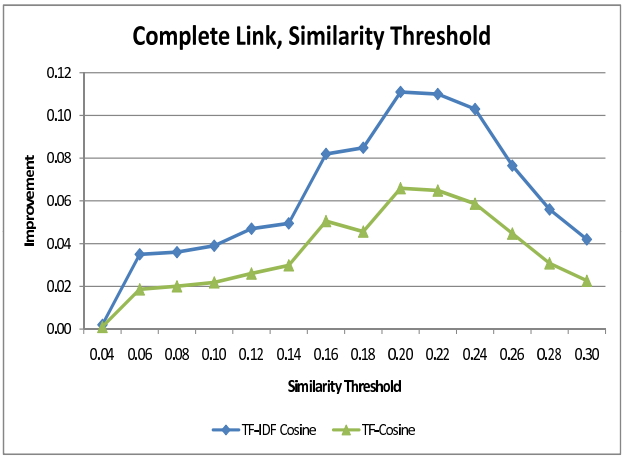

Figure 7: The effect of *similarity threshold* in maximal Complete Link clustering on the personalization algorithm.

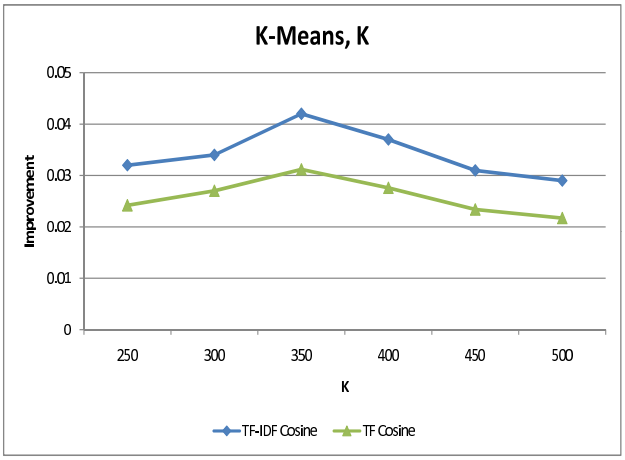

Figure 8: The effect of k in k-means Clustering on the personalization algorithm.

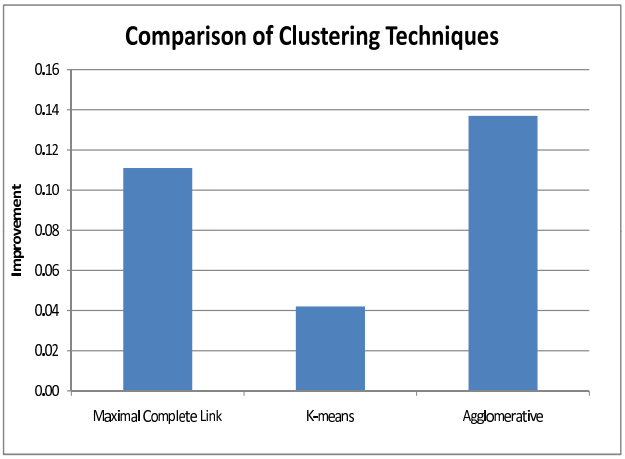

Figure 9: Comparison of the three techniques.

can be introduced and the personalization routine may not be able to improve performance. For these experiments the optimum value for the *generalization level* is 8 as shown in Figure 6. This is also the default value when testing other parameters.

The *generalization level* also has important ramifications related to the user's motivation. If the user is searching for something specific, a low *generalization level* may be chosen, focusing on clusters more directly related to the selected tag. However, if the user is browsing through the folksonomy, a higher *generalization level* may be appropriate. It may increase serendipity and introduce topics unknown but nevertheless interesting to the user. For example, users selecting the tag "chess" are likely to be interested in chess related resource. However, by increasing the *generalization level*, serendipitous discovery of other topics such as brain teasers or backgammon may be included.

If the *generalization level* is set very high, the algorithm will behave like a standard hierarchical clustering algorithm, returning all clusters below the *division coefficient*. In this case the performance of the personalization drops precipitously, underscoring the importance of the modifications to the algorithm.

The proposed personalization algorithm is independent from the method used to generate the tag clusters. In order to judge the relative benefit of the modified hierarchical clustering approach, clusters generated with maximal Complete Link and k-means clustering was also tested.

The *similarity threshold* for complete link has a strong impact on the effectiveness of the personalization routine as shown in Figure 7. If the similarity between two tags meets the *similarity threshold* they are considered connected; otherwise they are considered unconnected. If the threshold is set too low, links are generated between tags based upon a very weak relationship resulting in large clusters. On the other hand, setting the threshold too high, can remove links between tags that are in fact quite similar, resulting in a loss of valuable information, perhaps even resulting in numerous singleton clusters that have little added utility to the personalization method. Through empirical evaluation we found the ideal value for the similarity threshold to be 0.2, resulting in an improvement of .112.

We also investigated k-means clustering. It is an efficient clustering algorithm; the computational time it requires is much less than either hierarchical clustering or complete link clustering. It has only one parameter to tune, k, the predetermined number of clusters to be generated. If too many clusters are generated, a topic may be separated into many clusters. Alternatively, too few clusters can result in clusters covering multiple topics. In these experiments the optimum value for k was found to be approximately 350 as shown in Figure 8.

In sum, the evidence demonstrates that tag clusters can serve as effective intermediaries between users and resources thereby facilitating personalization. The personalization technique using maximal complete link clusters demonstrated promising results. It is particularly useful when dealing with ambiguous tags. By focusing on tags

specifically related to the user's selected tag, hierarchical clustering improved personalization even further. The worst of the three methods was k-means clustering, with a maximum improvement of only about .042 when compared to .112 and .137 as seen in Figure 9. Moreover, the standard deviation of the K-means clustering was .37. For maximal complete link the standard deviation was .30. It was .26 for hierarchical agglomerative clustering. Not only did hierarchical clustering prove to offer the most benefit to the personalization algorithm it was also the most reliable.

The poor results of k-means clustering can be attributed to its inability to identify innocuous tags. Both maximal Complete Link clustering and hierarchical agglomerative clustering use a similarity threshold to determine when to combine a tag in a cluster. As a result, many tags are clustered individually since they have little or no similarity to other tags. These singleton clusters of innocuous tags have little effect on the personalization algorithm. However, k-means clustering requires an input for the parameter k, and will put every tag into one of the clusters. These innocuous tags muddy the clusters and hinder the identification of the topic represented by the cluster. Without the ability to prune innocuous tags, clusters generated by k-means are ill suited for the proposed personalization algorithm.

Another drawback from k-means clustering is that ambiguous tags can pull unrelated tags together. For example, the tag "eve" can pull religious tags into the cluster as well as tags concerning online role-playing games such as EVE Online. Consequently, such a cluster can do little to alleviate the problem of tag ambiguity.

The strength of maximal Complete Link clustering lies in its ability to generate overlapping clusters, a trait well suited to the personalization algorithm. Ambiguous tags can be members of multiple clusters representing different interests. The tag "apple" for example can be a member of a cluster concerning the company, the fruit or New York City. A user that annotated a resource with an ambiguous tag will have some measure of similarity to all clusters containing the tag. The user profile can be used to disambiguate the intended meaning of the user, by measuring the relative interest of the user to each cluster.

The modified hierarchical clustering offers a level of customization not offered by the other two techniques. The parameter *step* effects the granularity of the hierarchy, while the *division coefficient* offers a means to select the cluster specificity. Moreover, the modifications to the algorithm coupled with the *generalization level* allow the algorithm to incorporate clusters strongly related to the user's selected tag or take a broader view of the hierarchy thereby promoting serendipitous discovery.

Conclusions and Future Work

In this work, we have proposed a method for personalizing search and navigation in folksonomies based on three clustering techniques. Tag clusters are used to bridge the gap between users and resources, offering a means to infer the user's interest in the resource. Standard search results based on cosine similarity are reordered using the relevance of the resource to the user. The clustering techniques are independent of the personalization algorithm, but the quality of the tag clusters greatly affects the performance of the personalization technique.

Several clustering approaches were investigated along with their corresponding parameters. By using clusters directly related to the selected tag, the modified hierarchical agglomerative clustering proved superior when compared to maximal complete link and k-means clustering. It offers not only better improvement in our experimental results, but provides more flexibility. The success of maximal complete Link clustering can be attributed in part to its ability generate overlapping clusters. Hence, ambiguous tags can be member of several clusters, and a user's intended meaning can be derived through the user profile. However, it is a well known NP-complete problem, and though the folksonomy is extremely sparse, this method may not scale well to larger samples. The k-means clustering algorithm, on the other hand, while very efficient, was not as successful in improving the personalized search results.

In the future, we plan to conduct work in several directions. We will continue to investigate different clustering approaches, including fuzzy versions of k-means algorithm. We plan to investigate potential improvements to the personalization algorithm itself. Alternative measures can be used to judge the relation of the users to the tag clusters, as well as the relation of the resources to the tag clusters. Also, the algorithm can be modified to support multi-tag queries.

Tag clusters can be used for other purposes, such as recommending tags or even users. Clusters of resource or can be used improving navigation in collaborative tagging systems, or possibly clusters of users. Other data mining and machine learning techniques can be used to overcome improve the user experience in folksonomies.

Acknowledgments

This work was supported in part by the National Science Foundation Cyber Trust program under Grant IIS-0430303 and a grant from the Department of Education, Graduate Assistance in the Area of National Need, P200A070536.

References

Augutson, J., and Minker, J. 1970. An Analysis of Some Graph Theoretical Cluster Techniques. *Journal of the Association for Computing Machinery* 17(4):571–588.

Begelman, G.; Keller, P.; and Smadja, F. 2006. Automated Tag Clustering: Improving search and exploration in the tag space. *Proceedings of the Collaborative Web Tagging Workshop at WWW* 6.

Chen, H., and Dumais, S. 2000. Bringing order to the Web: automatically categorizing search results. *Proceedings of the SIGCHI conference on Human factors in computing systems* 145–152.

Choy, S., and Lui, A. 2006. Web Information Retrieval in Collaborative Tagging Systems. *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence* 352–355.

Gower, J., and Ross, G. 1969. Minimum Spanning Trees and Single Linkage Cluster Analysis. *Applied Statistics* 18(1):54–64.

Hammond, T.; Hannay, T.; Lund, B.; and Scott, J. 2005. Social Bookmarking Tools (I). *D-Lib Magazine* 11(4):1082–9873.

Hayes, C., and Avesani, P. 2007. Using tags and clustering to identify topic-relevant blogs. *International Conference on Weblogs and Social Media*.

Heymann, P., and Garcia-Molina, H. April 2006. Collaborative Creation of Communal Hierarchical Taxonomies in Social Tagging Systems. Technical report, Technical Report 2006-10, Computer Science Department.

Johnson, D. 1973. Approximation algorithms for combinatorial problems. *Proceedings of the fifth annual ACM symposium on Theory of computing* 38–49.

MacQueen, J. 1967. Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability* 1(281-297):14.

Mathes, A. 2004. Folksonomies-Cooperative Classification and Communication Through Shared Metadata. *Computer Mediated Communication, LIS590CMC (Doctoral Seminar), Graduate School of Library and Information Science, University of Illinois Urbana-Champaign, December*.

Mika, P. 2007. Ontologies are us: A unified model of social networks and semantics. *Web Semantics: Science, Services and Agents on the World Wide Web* 5(1):5–15.

Millen, D.; Feinberg, J.; and Kerr, B. 2006. Dogear: Social bookmarking in the enterprise. *Proceedings of the Special Interest Group on Computer-Human Interaction conference on Human Factors in computing systems* 111–120.

Niwa, S.; Doi, T.; and Honiden, S. 2006. Web Page Recommender System based on Folksonomy Mining for ITNG06 Submissions. *Proceedings of the Third International Conference on Information Technology: New Generations (ITNG'06)-Volume 00* 388–393.

Salton, G., and Buckley, C. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing and Management: an International Journal* 24(5):513–523.

Salton, G.; Wong, A.; and Yang, C. 1975. A vector space model for automatic indexing. *Communications of the ACM* 18(11):613–620.

Teevan, J.; Dumais, S.; and Horvitz, E. 2007. Characterizing the value of personalizing search. *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval* 757–758.

Van Rijsbergen, C. 1979. *Information Retrieval*. Butterworth-Heinemann Newton, MA, USA.

Voorhees, E. 1999. The TREC-8 Question Answering Track Report. *Proceedings of TREC* 8:77–82.

Wu, X.; Zhang, L.; and Yu, Y. 2006. Exploring social annotations for the semantic web. *Proceedings of the 15th international conference on World Wide Web* 417–426.

Xu, Z.; Fu, Y.; Mao, J.; and Su, D. 2006. Towards the semantic web: Collaborative tag suggestions. *Collaborative Web Tagging Workshop at WWW2006, Edinburgh, Scotland, May*.

Yan, R.; Natsev, A.; and Campbell, M. 2007. An efficient manual image annotation approach based on tagging and browsing. *Workshop on multimedia information retrieval on The many faces of multimedia semantics* 13–20.