The Impact of Ambiguity and Redundancy on Tag Recommendation in Folksonomies

Jonathan Gemmell, Maryam Ramezani, Thomas Schimoler
Laura Christiansen, Bamshad Mobasher
Center for Web Intelligence
School of Computing, DePaul University
Chicago, Illinois, USA
{jgemmell, mramezani, tschimo1, lchris10, mobasher}@cdm.depaul.edu

ABSTRACT

Collaborative tagging applications have become a popular tool allowing Internet users to manage online resources with tags. Most collaborative tagging applications permit unsupervised tagging resulting in tag ambiguity in which a single tag has many different meanings and tag redundancy in which several tags have the same meaning. Common metrics for evaluating tag recommenders may overestimate the utility of ambiguous tags or ignore the appropriateness of redundant tags. Ambiguity and redundancy may even burden the user with additional effort by requiring them to clarify an annotation or forcing them to distinguish between highly related items. In this paper we demonstrate that ambiguity and redundancy impede the evaluation and performance of tag recommenders. Five tag recommendation strategies based on popularity, collaborative filtering and link analysis are explored. We use a cluster-based approach to define ambiguity and redundancy and provide extensive evaluation on three real world datasets.

1. INTRODUCTION

Collaborative tagging applications, also known as folksonomies [18], have emerged as a powerful trend allowing Internet users to annotate, share and explore online resources. Delicious[1] supports users as they bookmark URLs. Citeulike[2] enable researchers to manage scholarly references. Bibsonomy[3] allows users to tag both. Still other collaborative tagging systems specialize in music, photos and blogs.

At the core of collaborative tagging systems is the post: a user describes a resource with a set of tags. Taken in isolation, an individual post allows a user to organize web resources for later use: resources can be easily sorted, aggregated and retrieved. Taken as a whole the sum of many posts results in a complex network of interrelated users, resources and tags that is useful in its own right or for data mining techniques that support the user's activities.

[1]delicious.com
[2]citeulike.org
[3]bibsonomy.org

RecSys'09, October 22–25, 2008, New York City, New York.

Despite the many benefits offered by folksonomies, they also present unique challenges. Most collaborative tagging applications allow the user to describe a resource with any tag they choose. As a result they contain numerous ambiguous and redundant tags. An ambiguous tag has multiple meanings: "apple" may refer to the fruit or the company. Redundant tags share a common meaning: "America" and "USA" confer the same idea.

These tags make it difficult to judge the effectiveness of tag recommendation algorithms which suggest tags for a user during the annotation process. Standard evaluation approaches will often view recommended ambiguous tags as hits when they appear in a holdout set even if the underlying meaning of the recommended tag was different than the context in which it appears in the holdout set. Such recommendations, therefore, while appearing to be effective in the evaluation process, in reality may mislead users as they search for relevant resources.

Redundancy can hamper the effort to judge recommendations as well, but from the opposite perspective. A recommended tag may be counted as a miss even though it is synonymous to a tag in the holdout set. An example may be when the holdout set for a test user contains the tag "java" while the recommendation set contains "Java." Therefore, redundancy may mask the true effectiveness of the recommendation algorithm: while from the user's perspective "Java" is a good recommendation, in the evaluation process it would appear as incorrect.

Our goal in this work is to determine the impact of these two phenomena on the effectiveness of tag recommendation. We employ a cluster-based approach to define and measure ambiguity and redundancy. We cluster both resources and tags into highly cohesive partitions based on co-occurrence. A tag is considered ambiguous if several resources from different clusters have been annotated with it. Tags from the same tag cluster are considered redundant. We chose the cluster-based approach over a variety of semantic and linguistic approaches because it provides a more general and language independent method for defining ambiguity and redundancy. We define metrics, based on the resulting clusters, for measuring the degree of ambiguity for a tag, and the level of redundancy for pairs of tags. We provide extensive evaluation on three real world Folksonomies to determine the impact of ambiguity and redundancy across several common tag recommendation algorithms as well as across data sets.

The rest of this paper is organized as follows. In Section 2 we detail previous work related to tag recommendation in Folksonomies and efforts in identifying and quantifying ambiguity and redundancy. In Section 3 we briefly describe our clustering approach and then detail our measures for ambiguity and redundancy. Section 4 reviews five tag recommendation strategies employed in our

evaluation. Our methodology, datasets and experimental results are offered in Section 5. Finally, we conclude the paper with directions for future work.

2. BACKGROUND AND RELATED WORK

The term "folksonomy" was first coined in [21], a portmanteau of "folk" and "ontology". Folksonomies permit users to annotate online resources with tags. These tags can serve multiple functions: convey ownership ("jon"), describe the resource ("article"), describe the characteristics of the resource ("folksonomies"), provide subjective commentary ("cool"), or help organize the resource ("toread") [8].

A folksonomy can be described as a four-tuple: a set of users, U; a set of resources, R; a set of tags, T; and a set of annotations, A. We denote the data in the folksonomy as D and define it as: $D = \langle U, R, T, A \rangle$.

The annotations, A, are represented as a set of triples containing a user, tag and resource defined as: $A \subseteq \{\langle u, r, t \rangle : u \in U, r \in R, t \in T\}$.

A folksonomy can therefore be viewed as a tripartite hyper-graph [19] with users, tags, and resources represented as nodes and the annotations represented as hyper-edges connecting one user, one tag and one resource.

Many authors have attempted to exploit this data structure in order to recommend resources, tags or even other users. In [10] the authors proposed an adaptation of link analysis to the folksonomy data structure for resource recommendation. They have called this technique *Folkrank* since it computes a Pagerank [3] vector from the tripartite graph. In [11] Folkrank is used for tag recommendation. While it suffers from extreme computational costs, it has proven to be one of the most effective tag recommenders; as such we have included it in our evaluation.

In [17, 11] traditional collaborative filtering algorithms were extended for tag recommendation in folksonomies. In [15] the authors personalized tag recommendation by considering the user profile as well as the tags most often applied to the resource being annotated. Because if its simplicity, popularity and effectiveness we analyze this technique as well.

Ambiguity is a well known problem in information retrieval and has been identified as a problem in folksonomies as early as [18]. WordNet has been used to identify ambiguous tags and disambiguate them by using synonyms [14]. Folksonomy searches are expanded with ontologies in [12] to solve the ambiguity in tagging systems. Clustering was used to measure ambiguity in [24]. They focus on the network analysis techniques to discover clusters of nodes in networks. In [13] multi-dimensional scaling is used for co-word clustering to visualize the relationships between tags. We also presume that tags and resources can be aggregate into tightly related clusters.

Entropy as a measure of tag ambiguity has been proposed in [25, 23]. They used a probabilistic generative model for data co-occurrence to determine ambiguous tags. The tagging space is assume to cover different categories and the tag membership of each category is estimated via the EM algorithm. Our measure of ambiguity uses a similar approach. We cluster resources and use the distribution of tags across the clusters to measure their ambiguity.

Redundancy in a folksonomy is due largely to the ability of users to tag resources irrespective of a strict taxonomy. In [8] two types of redundancy are identified: structural and synonymic. In [22] structural redundancy is explained by stemming to remove suffixes, removing stop words, comparing tags for differences of only one character or identifying compound tags. Synonymic redundancy is evaluated in [1] by using WordNet to determine synonyms. Clustering has also been utilized to identify redundancy. In [6, 7] agglomerative clustering is used to identify similar tags. Since stemming and lexical databases are ill suited to deal with the inherent chaos found in tags we rely on clustering to aggregated tags into single topic clusters.

3. MEASURING AMBIGUITY AND REDUNDANCY

In this paper we utilize a cluster based definition of ambiguity and redundancy. Resources are modeled as a vector over the set of tags. In calculating the vector weights, a variety of measures can be used: recency, adjacency, or frequency. In this work we rely on frequency. The *term frequency* for a resource tag pair is the number of times the resource has been annotated with the tag. We define *tf* as: $tf(r,t) = |\{a = \langle u, r, t \rangle \in A : u \in U\}|$.

Several techniques exist to calculate the similarity between vectors such as Jaccard similarity, Pearson correlation, or cosine similarity. We rely on cosine similarity [20]. Similarity between tags, modeled as a vector over the resource space, can be defined analogously.

While the approach for measuring ambiguity and redundancy is independent from any specific clustering method, because of its speed, simplicity and popularity we rely on K-Means clustering [5, 16]. Resources are randomly partitioned into k initial sets. Ascertaining the ideal value k is a difficult problem. In this work we use both subjective evidence (visually inspecting the clustering) and Hubbert's correlation with distance matrix [4] defined as:

$$h(k) = (1/M) \sum_{i}^{k} \sum_{j=i+1}^{k} C(t_i, t_j) S(t_i, t_j) \quad (1)$$

where $M = n(n-1)/2$, the number of tag pairs in the folksonomy; $C(t_i, t_j)$ is 1 if the two tags are in the same cluster; and $S(t_i, t_j)$ is the similarity between the two tags.

Several values of k are evaluated using $h(k)$, and k is chosen such that tightly connected tag clusters are not further separated for larger values of k.

3.1 Ambiguity

Ambiguous tags have multiple meanings. A tag may have different word senses; "apple" can refer to the company or to the fruit. Names may also result in ambiguity; "paris" might mean the city or the celebrity. Subjective tags such as "cool" can result in ambiguity since different users have contradictory notions of what constitutes cool. Finally, overly vague tags such as "tool" can mean gardening implements to some or software packages to others.

Ambiguous tags can impede users as they navigate the system or burden the user with unwanted recommendations. At a systems level ambiguous tags can introduce erroneous features into the user profile. While recommenders are often judged by their ability to predict items occurring in a holdout set, the quality of a tag recommender may be underestimated by traditional metrics if it routinely passes up these tags in order to recommend tags with greater information value. Moreover, evaluation metrics may overvalue a recommender that proposes ambiguous tags despite their lack of specificity.

We define a tag as ambiguous if it has been applied to several resources from among different resource clusters. Assuming that the clustering algorithm has effectively aggregated similar resources and separated resources with little similarity, then a tag which has been annotated to resources of many clusters can be assumed to be more ambiguous than a tag which has been narrowly applied.

Given a cluster of resources, $c \in C_r$, we may then define the cluster frequency, *cf*, of a tag as the sum of the term frequencies over the resources in the cluster:

$$cf(c,t) = \sum_{r \in c} tf(r,t) \tag{2}$$

We may then adapt entropy in order to compute the ambiguity of a tag, $a(t)$, as:

$$a(t) = -\sum_{c \in C_r} \left(\frac{cf(c,t)}{f(t)} \cdot log \frac{cf(c,t)}{f(t)} \right) \tag{3}$$

where $f(t)$ is the frequency of the tag in the folksonomy.

The entropy of a tag reflects its ambiguity by revealing whether it is distributed across many resource clusters or if it is confined to only a few clusters. We may further define the ambiguity of a recommendation set as the average ambiguity of its tags.

$$A(T_r) = \frac{\sum_{t \in T_r} a(t)}{|T_r|} \tag{4}$$

3.2 Redundancy

Because users may annotate resources with any tag they choose, folksonomies are laden with redundant tags that share a common meaning. Syntactic variance such as "blogs" or "blogging" can cause redundancy. Case ("java" or "Java"), spelling ("gray" or "grey"), and multilinguism ("Photo" or "Foto") may also result in redundancy. The use of non-alphanumeric characters, abbreviations, acronyms and deliberate idiosyncratic tagging are other sources of redundancy.

Redundant tags make it difficult to the judge the quality of recommendations. Traditionally a holdout set is employed to rate an algorithm's effectiveness. However if the utility metric defines success as the ability to exactly match the holdout set, the usefulness of the recommender can be undervalued. Redundant tags that clearly reflect the user's intent in the holdout set should be counted as matches; this more accurately reflects the benefit of the recommendation to the user.

For example, consider the case in which "RecSys" is in the holdout set and "rec_sys" is suggested. Standard evaluation techniques would count this as a miss, no better than if it had recommended a completely irrelevant tag. Instead, if the two tags are known to be redundant, the redundancy aware metric would count the suggested tag as a hit.

In order to detect redundancy we rely on clusters of tags. Two tags are considered redundant if they are members of the same cluster. When evaluating tag recommenders we use recall and precision as well as a redundancy aware versions that rely on clusters of tags.

Recall is a common metric of recommendation algorithms that measures coverage. It measures the percentage of items in the holdout set, T_h, that appear in the recommendation set T_r. It is defined as: $r = (|T_h \cap T_r|)/|T_h|$. Precision is another common metric that measures specificity and is defined as: $p = (|T_h \cap T_r|)/|T_r|$

Redundancy aware recall and precision assume a set of tag clusters. A recommended tag is considered a hit if it or a tag in its cluster also appears in the holdout set:

$$r_r = \frac{\sum_{\mathrm{i} \in \mathrm{T_r}} \mathrm{R(i, T_h)}}{|\mathrm{T_h}|} \tag{5}$$

where $R(i, T_h)$ is defined as 1 if i or one of its redundant tags appears in T_h, and 0 otherwise. Redundancy aware precision is similarly defined:

$$p_r = \frac{\sum_{i \in T_r} R(i, T_h)}{|T_r|} \tag{6}$$

As in standard recall and precision, the redundancy aware metrics will fall between 0 and 1.

In order to calculate the redundancy of a recommendation set, each tag-tag pair in the set is evaluated as to whether or not they appear in the same cluster:

$$R(T_r) = \frac{1}{N} \sum_{i}^{|T_r|} \sum_{j=i+1}^{|T_r|} C(t_i, t_j) \tag{7}$$

where N is the number of tag pairs in T_r and $C(t_i, t_j)$ is 1 if the two tags share a cluster, and 0 otherwise.

4. TAG RECOMMENDATION

Here we review several common recommendation techniques which we employ in our evaluation. Recommendation in folksonomies may include the suggestion of tags during the annotation process, new resources to users as they navigate the system or even other users that share common interests. In this paper we focus on tag recommendation. We consider techniques based on popularity, collaborative filtering and linkage analysis.

In traditional recommendation algorithms the input is often a user, u, and the output is a set of items, I. Tag recommendation in folksonomies differs in that the input is both a user, u, and a resource, r. The output remains a set of items, in this case a recommended set of tags, T_r. For each recommendation approach we define $\tau(u, r, t)$ to be the relevance of tag, t, for the user-resource pair. A recommender will return the top n tags with the highest τ.

4.1 Popularity Based Approaches

Perhaps the simplest recommendation strategy is merely to recommend the most commonly used tags in the folksonomy. Alternatively, given a user-resource pair a recommender may ignore the user and recommend the most popular tags for that particular resource. This strategy is strictly resource dependent and does not take into account the tagging habits of the user. We define τ for resource based popularity, pop_r, recommendations as:

$$\tau(u, r, t) = \frac{|\{a = \langle u, r, t \rangle \in A : u \in U\}|}{|\{a = \langle u, r, t \rangle \in A : u \in U, t \in T\}|} \tag{8}$$

In a similar fashion a recommender may ignore the resource and recommend the most popular tags for that particular user. While such an algorithm would include tags frequently applied by the user, it does not consider the resource information and may recommend tags irrelevant to the current resource. We define τ for user based popularity, pop_u, recommendations as:

$$\tau(u, r, t) = \frac{|\{a = \langle u, r, t \rangle \in A : r \in R\}|}{|\{a = \langle u, r, t \rangle \in A : u \in U, t \in T\}|} \tag{9}$$

4.2 K-Nearest Neighbor

User Based K-Nearest Neighbor is a commonly used recommendation algorithm in Information Retrieval that can be modified for tag recommendation in folksonomies. Traditionally it finds a set of users similar to a query user. From these neighbors a set of recommended items is constructed.

We can modify this approach by ignoring neighbors that have not tagged the query resource. Once a neighborhood of similar users

has been discovered, the algorithm considers only those tags that have been applied to the query resource and calculates a relevance for each tag, τ, as the average similarity of the neighbors that have applied the tag.

Algorithm 1 K-Nearest Neighbor Modified for Folksonomies

Require: u, the query user; r, the query resource; n, the number of tags to recommend

1: **for** each $u \in U$ that has annotated r_q **do**
2: s_u = sim(u, u_q)
3: **end for**
4: Let N be the k nearest neighbors to u_q
5: **for** each $u \in N$ **do**
6: **for** each t that u applied to r_q **do**
7: $\tau = \tau + s_u/k$
8: **end for**
9: **end for**
10: Return T_r, the top n tags sorted by τ;

Users may be modeled in a myriad of ways; here we model each user as a vector over the set of tags We again rely on *term frequency* to calculate the weights in the vector using either tag counts or resource counts and use similarity to define the similarity, $sim(u_1, u_2)$, between users. We call these methods KNN_{ut} and KNN_{ur} respectively

4.3 Folkrank

Folkrank was proposed in [10]. It computes a Pagerank vector from the tripartite graph of the folksonomy. This graph is generated by regarding $U \cup R \cup T$ as the set of vertices. Edges are defined by the three two-dimensional projections of the hypergraph.

If we regard the adjacency matrix of this graph, W, (normalized to be column-stochastic), a damping factor, d, and a preference vector, p, then we iteratively compute the Pagerank vector, w, in the usual manner: $w = dAw + (1 - d)p$.

However due to the symmetry inherent in the graph, this basic Pagerank may focus to heavily on the most popular elements. The Folkrank vector is taken as a difference between two computations of Pagerank: one with and one without a preference vector. Tag recommendations are generated by biasing the preference vector towards the query user and resource [11]. These elements are given a substantial weight while all other elements have uniformly small weights.

We include this method as a benchmark as it has been shown to be an effective method of generating tag recommendations. However, it imposes steep computational costs.

5. EXPERIMENTAL EVALUATION

In this section we discuss our datasets. We then discuss our experimental methodology. Since the clustering of tags and resources is a fundamental step in our ambiguity and redundancy metrics we provide both empirical and anecdotal evidence of the quality of the clusters. We then provide two separate collections of experiments: the first on ambiguity, the second on redundancy.

5.1 Datasets

We have chosen three datasets for our experiments: Delicious, Bibsonomy and Citeulike. In order to reduce noise and focus on the denser portion of the dataset a P-core was taken such that each user, resource and tag appear in at least p posts as in [2, 11]. Table 1 shows the distribution of the datasets.

Delicious				
	All	-1% amb.	-2% amb.	-3% amb.
|U|	18,105	17,985	17,982	17,981
|R|	42,646	42,646	42,646	42,646
|T|	13,053	12,923	12,792	12,662
|P|	2,309,427	2,104,072	2,017,615	1,954,841
|A|	8,815,545	6,123,705	5,225,134	4,716,228
Bibsonomy				
	All	-1% amb.	-2% amb.	-3% amb.
|U|	346	346	345	345
|R|	1,639	1,639	1,639	1,639
|T|	1,652	1,636	1,619	1,603
|P|	13,082	12,648	12,278	12,104
|A|	48,039	38,451	35,225	33,270
Citeulike				
	All	-1% amb.	-2% amb.	-3% amb.
|U|	2,051	2,047	2,046	2,044
|R|	5,376	5,376	5,376	5,374
|T|	3,343	3,310	3,277	3,243
|P|	42,278	40,608	39,631	38,454
|A|	105,873	88,720	82,408	76,160

Table 1: The number of users, resources, tags, posts and annotations for the datasets before and after ambiguous tags are removed.

Delicious is a popular Web site in which users annotate URLs. On 10/19/2008, 198 of the most popular tags were taken from the user interface. For each of these tags the 2,000 most recent annotations including the contributors of the annotations were collected. This resulted in 99,864 distinct usernames. For each user the social network was explored recursively resulting in a total of 524,790 usernames.

From 10/20/2008 to 12/15/2008 the complete profiles of all users were collected. Each user profile consisted of a collection of posts including the resource, tags and date of the original bookmark. The top 100 most prolific users were visually inspected; twelve were removed from the data because their post count was many orders of magnitude larger than other users and were suspected to be Webbots. Due to memory and time constraints, 10% of the user profiles was randomly selected. A p-core of 20 was taken from this dataset for experiments.

The Bibsonomy dataset was gathered on 1/1/2009 encompassing the entire system. This data set has been made available online by the system administrators [9]. They have pre-processed the data to remove anomalies. A 5-core was taken to reduce noise and increase density.

Citeulike is used by researchers to manage and discover scholarly references. The dataset is available to download. On 2/17/2009 the most recent snapshot was taken. The data contains anonymous user ids and posts for each user including resources, the date and time of the posting and the tags applied to the resource. A P-core of 5 was calculated.

5.2 Experimental Methodology

We employ the leave one post out methodology as described in [10]. One post from each user was placed in the testing set consisting of a user, u, a resource, r, and all the tags the user has applied to that resource. These tags, T_h, are analogous to the holdout set commonly used in Information Retrieval evaluation. The remaining posts are used to generate the recommendation models.

The tag recommendation algorithms accepts the user-resource pair and returns an ordered set of recommended tags, T_r. From the holdout set and recommendation set utility metrics were calculated.

Tag Cluster 1	Tag Cluster 2
collaborative_filtering	New
collaborativefiltering	York
filtering	brooklyn
recommendation	new-york
recommender	new_york
svd	newyork
taste	newyorkcity
	ny
	nyc
	overheard
	york
Tag Cluster 3	**Tag Cluster 4**
Folksonomy	Clustering
Tag	Cluster
Tagging	cluster
acts_as_taggable	clustering
folksonomia	failover
folksonomies	highavailability
folksonomy	hpc
tag	lvs
tagging	redundancy
tags	supercomputer
	terracotta
Resource Cluster 1	**Resource Cluster 2**
albumart.org	mapsforus.org
albumartwork.org	strangemaps.wordpress.com
allcdcovers.com	davidrumsey.com
cdcovers.cc	elbruz.org
coverdude.com	handmaps.org
findmycover.com	helmink.com
freecovers.net	hipkiss.org/cgi-bin/maps.pl
slothradio.com/covers	mcwetboy.net/maproom
	radicalcartography.net
	urbancartography.com

Table 2: Selected resource and tag clusters.

For each metric the average value was calculated across all test cases.

5.3 Clusters

The clustering algorithm is independent of the ambiguity and redundancy measures. Still, we assume that tags and resources can be aggregated into coherent clusters representing distinct topic areas. Support for that assumption is given in Table 2 where four tag clusters are presented.

Cluster 1 represents the idea of "Recommendation", while cluster 2 represents New York[4]. Other clusters show clearly recognizable categories. Clusters can capture misspellings, alternative spellings or multilinguism such as in cluster 3: "Folksonomy" versus "folksonomies." They can also capture tags that share a similar concept space: "recommendation" and "collaborativefiltering."

Cluster 4, on the other hand, shows how clustering can be effected by ambiguity. Two similar yet distinct senses of clustering have been aggregated: clustering algorithms and computational clusters[5].

Visual examination of two resource clusters also shows the effectiveness of aggregating similar items. In the example resources are clearly related either to album artwork or to maps.

Despite the apparent ability of the clustering approach to aggregate items into well defined cohesive clusters, an objective measure is still needed to select k. In Figure 1 Hubert's correlation with distance matrix is calculated for several values of k for tag and resource clusters in Delicious. When k is approximately 2500 we observe a knee in the progression. We have interpreted this to mean that tightly focused clusters are not broken up for larger values of k, and have consequently selected this value. In order to preserve space we do not here report all experiments for k on different datasets and for resource and tag clusters, but we observe a similar trend for all experiments and a have chosen k accordingly. For Bibsonomy we have chosen 500 for k in both tag and resource clusters and in Citeulike we have chosen 750 and 500 respectively.

[4]overheardinnewyork.com is a popular web blog.

[5]terracotta is an opensource project for computational clusters

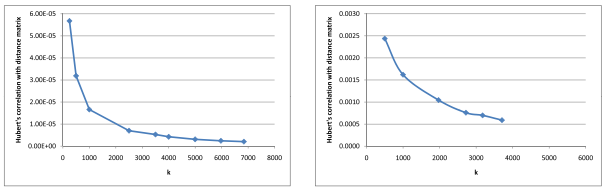

Figure 1: Evaluation of k for resource and tag clusters in Delicious using Hubert's correlation with distance matrix.

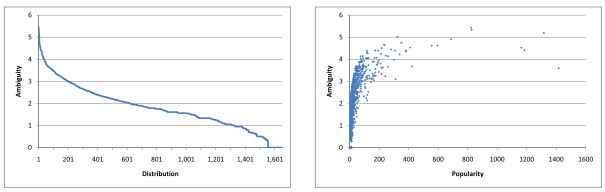

Figure 2: The distribution of ambiguity in Bibsonomy. Popularity versus ambiguity in Bibsonomy.

5.4 Impact of Ambiguity

Ambiguity can cause standard utility metrics to overestimate the effectiveness of tag recommenders by rewarding ambiguous recommendations even as they contribute to noise, confound the user experience, clutter the user profile and impose additional effort on the user. The effectiveness of the recommenders may even be underestimated when they avoid such tags, penalizing them for not suggesting the ambiguous tags found in the holdout set.

In Figure 2 we show the distribution of Ambiguity for all 1651 tags in the Bibsonomy dataset. A few tags are very ambiguous while the majority have a relatively low level of ambiguity. Figure 2 also shows the correlation between popularity and ambiguity. A very popular tag is likely to also be ambiguous perhaps because it can be applied in many different contexts, though ambiguous tags are not necessarily popular demonstrating that popularity alone is not a good predictor of ambiguity. The Delicious and Citeulike datasets show nearly identical trends.

Table 3 lists the top ten ambiguous tags in Delicious. "Bookmarks" and "imported" seem to be system tags that have been applied during an import operation and thus have been applied to several resources regardless of context. Subjective tags such as "cool," "interesting," and "useful" are ambiguous because different people find alternative subjects interesting. Consequently, subjective tags can be applied to any resource cluster. "Tools" and "tools" are ambiguous because they are overly vague, perhaps meaning hand tools to some and data mining tools to others. The tag "toread" is a functional tag and is ambiguous because users mark items to read across disparate topics. Five of the top ten ambiguous tags in

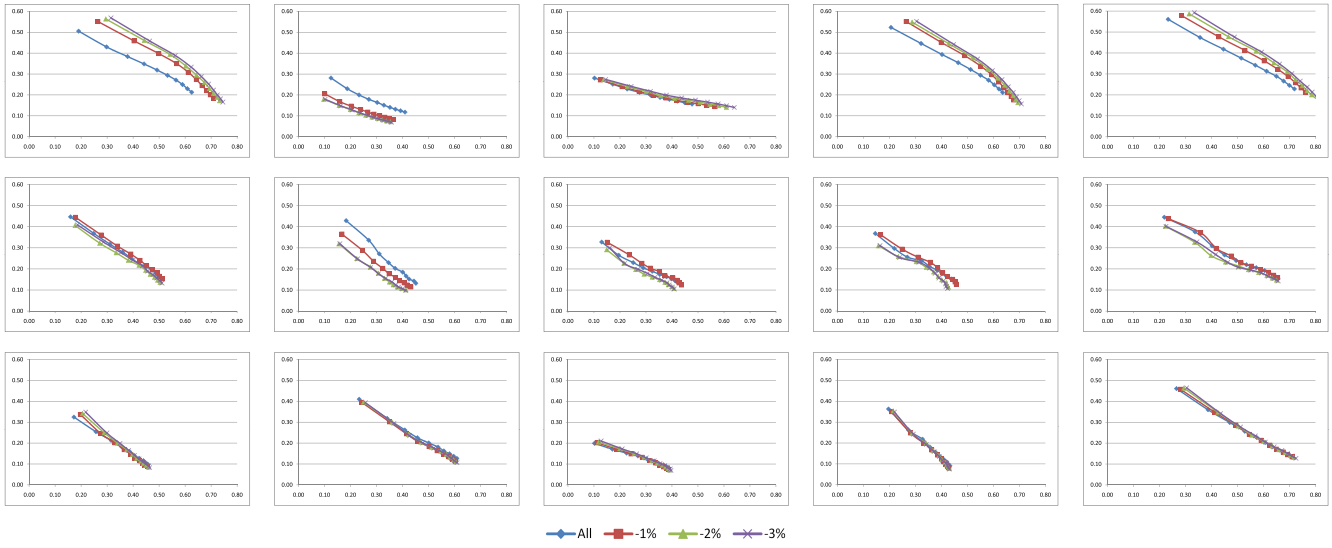

Figure 3: From top to bottom, experiments completed on Delicious, Bibsonomy and Citeulike. From left to right, most popular by resource, most popular by user, K-nearest neighbor - resource model, K-nearest neighbor - tag model and Folkrank experiments. All figures show recall on the x-axis from 0.0 to 0.8 and precision on the y-axis from 0.0 to 0.6. Each line shows the recall-precision value for recommendation sets of size 1 through 10. The complete datasets, as well as datasets with 1, 2 and 3 percent of the top ambiguous tags removed are shown with diamonds, squares, triangles and crosses respectively.

Tag	Ambiguity	Frequency
Bookmarks	6.936	7,835
imported	6.885	21,770
cool	6.802	42,392
interesting	6.783	16,413
toread	6.683	19,637
tools	6.680	182,629
reference	6.616	135,471
useful	6.611	12,157
Tools	6.573	9,946
web	6.568	136,953

Table 3: Top 10 ambiguous tags in Delicious along with their ambiguity score and frequency.

Delicious also appear in the top tags for Bibsonomy. This is likely due to the fact that both systems allow the user to annotate URLs and therefore share common a domain, though Bibsonomy users also annotate scholarly articles. The most ambiguous tags in Citeulike appear to be overly vague tags such as "model," "theory," and "software." Within the scope of scientific publication these tags are quite indeterminate.

In order to measure the impact of ambiguous tags across the folksonomies we remove the top one, two and three percent of ambiguous tags from the datasets. Table 1 shows the impact of the datasets with the ambiguous tags removed. Rarely is a user or resource completely removed through this process. Moreover the number of posts (a user, resource and *all* tags applied to that resource) is reduced marginally while the number of triples (user-resource-tag tuples) is dramatically reduced. This shows that while users routinely apply ambiguous tags, they rarely annotate a resource with ambiguous tags alone.

To ascertain the impact of ambiguity on the tag recommendation strategies we perform the five recommendation techniques with the original datasets as well as datasets generated by removing ambiguous tags. For the k-nearest neighbor recommendation strategy we tuned k on the original datasets. For Delicious we found 10 to be optimal for both KNN_{ur} and KNN_{ut}. In Bibsonomy we used 5 for both techniques, and in Citeulike we used 5 and 10 respectively. For Folkrank we verified the optimal value of 0.3 for d as in [11].

We then measure the effectiveness of the tag recommendation strategies with recall and precision as shown in Figure 3. The left most column in the figure shows the impact of removing ambiguous tags on the most popular by resource strategy, pop_r. Since pop_r recommends tags based upon their popularity for a particular resource, the recommended tags tend to be quite specific in describing the resource. The removal of ambiguous tags from the data set, therefore will either result in little change in recommendation effectiveness or (in the case of broad folksonomies, such as Delicious) may actually reduce the noise and result in higher recommendation accuracy.

We observe the opposite effect with the most popular by user technique, pop_u applied to the Delicious dataset. In this approach, the recommended tags (those that the user has often used across resources) are much more likely to be inheretly ambiguous. A user for example that often annotates resources with "toread", can be presumed to continue this behavior regardless of the characteristics of the resource. Therefore, removing such ambiguous tags from the data may actually result in reduced precision (from the perspective of the evaluation methodology). Bibsonomy shows similar results while Citeulike reveals little change. The behavior in Citeulike may be due to the fact that users tag a much more spefcific set of resources (scientific articles) based on their interest in specific areas. Thus, most popular tags by a user tend to be less ambiguous than in a broad folksonomy such as Delcious.

These two examples demonstrate the difficulty in evaluating and comparing the true effectiveness of tag recommendation strategies. Recommendation by pop_r is often contextually appropriate but ambiguous tags in the user's holdout set masks the true value of this technique. On the other hand, ambiguous tags cause recall and pre-

cision to overrate the quality of pop_u: the recommended tags offer little utility and does not provide the user with new or contextually significant options.

As would be expected, more robust recommenders such as K-nearest neighbor are more resilient to ambiguity. As shown in Table 1 the number of resources changes very little despite the removal of ambiguous tags. Consequently when users are modeled as vectors over the set of resources such as in KNN_{ur} we see very little change in the performance across the reduced datasets.

However, in KNN_{ut} the inclusion of ambiguous tags may obfuscate the the user model and imply false similarities among users. The Delicious experiment shows how ambiguous tags may have muddied the user profile. Removing the ambiguous tags from the data, and thus the holdout set, the recommender is able to focus on more context oriented annotations. In contrast, the Bibsonomy and Citeulike experiments reveal little change in KNN_{ut} when removing ambiguous tags, likely because the average ambiguity of tags in a post is far less in Bibsonomy and Citeulike (2.24 and 1.91, respectively) compared to Delicious (3.78). Here we see that ambiguity not only plays a role when comparing recommendation techniques but when comparing datasets as well.

Folkrank behaves in a similar manner. In Delicious, where ambiguity is more common, the removal of ambiguous tags makes it easier to model users, in this case through linkage analysis. Experiments on Citeulike and Bibsonomy show little change.

Looking across the datasets rather than over the recommendation techniques reveals additional insights. In general Citeulike exhibits little change regardless of the removal of ambiguous tags. This is not surprising since it has the lowest average ambiguity over its posts (1.91), most likely because it is focused on scholarly journals and its members have an added incentive to organize their resources with more focused tags.

Delicious members on the other hand are often casual users that annotate a broad range of topics. Consequently it has the largest average ambiguity among posts (3.78). We therefore witness dramatic changes in this dataset. Bibsonomy, a collaborative tagging system that includes both URLs and journal articles, has an average ambiguity over posts of 2.24, falling between the other two.

5.5 Impact of Redundancy

Tag redundancy may also mask the true effectiveness of the recommendation algorithm: while from the user perspective a tag may be a good recommendation, in the evaluation process it might appear as ineffective if it is very similar but not identical to the holdout tag. We can measure the impact of redundancy by comparing recommendation accuracy in the standard evaluation framework to the redundacy-aware framework defined in Section 3.2 .

Figure 4 shows the most popular by resource recommendation strategy, pop_r, on the Delicious dataset and the Folkrank, $folk$, algorithm on the Citeulike dataset. Both techniques are evaluated with standard recall and precision as well as the redundancy-aware metrics that rely on clusters of tags. In both cases we have observed that redundant tags have been suggested, but were ignored by the standard methods.

For simplicity we define dif_r to be the average difference between standard recall and redundancy-aware recall across recommendation sets of size 1 through 10. In the precision case, dif_p is defined similarly. The results are provided in Table 4. In all strategies and datasets we observed a similar trend, but to different degrees. We found the smallest improvement in pop_u due to the fact that recommended tags are drawn from the target user's own profile, and because users tend to focus on a single tag from a redundant cluster rather than employing several redundant variations.

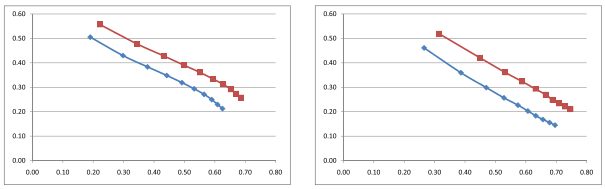

Figure 4: pop_r on Delicious and $folk$ on Citeulike. Recall is reported on the x-axis, precision on the y-axis. The lower line shows recall-precision for recommendation sets of size 1 through 10. The upper line shows the redundancy aware versions.

	Delicious		Bibsonomy		Citeulike	
	dif_r	dif_p	dif_r	dif_p	dif_r	dif_p
pop_r	0.056	0.044	0.040	0.037	0.083	0.068
pop_u	0.006	0.004	0.006	0.007	0.019	0.017
knn_{ur}	0.075	0.063	0.073	0.066	0.091	0.064
knn_{ut}	0.051	0.039	0.071	0.065	0.075	0.056
$folk$	0.047	0.044	0.060	0.074	0.056	0.064

Table 4: The difference in standard recall and precision and their redundancy aware counterparts.

The greatest difference is observed in knn_{ur} suggesting that standard evaluation techniques may underestimate its true effectiveness from the user perspective. The other recommendation strategies also show a marked difference in dif_r and dif_p suggesting that many recommenders have difficultly penetrating the noise created by redundant tags.

We also observed differences among the redundancy of recommendation sets as defined in Equation 7. Table 5 shows that pop_r suggests far more redundant tags that the other techniques. This is likely due to the fact that it solely focuses on the resource whose representation includes redundant variants of tags associated by many users to that resource. On the other hand pop_u, which looks only at the user, generates little redundancy. The other techniques possess moderate redundancy.

The quality of a recommender may be overvalued if the inclusion of several redundant tags imposes additional effort on the user. Users may be required to judge the nuances between similar tags or randomly choose one of several tags. Moreover they may be forced to hunt for a previously used tag from a collection of synomyms or similar tags.

6. CONCLUSIONS

Ambiguity can give a false impression of success when the recommended tags offer little utility. Such recommendations while appearing to be effective in the evaluation process may mislead users as they search for relevant resources. Recommenders that avoid ambiguous tags may be penalized by standard utility metrics for not promoting such tags. More robust algorithms such as K-nearest neighbor and Folkrank weather ambiguity better than their simpler counterparts. However when users are modeled with tags, ambiguity can pollute the user profile and impede the performance of the recommender. We have also discovered that ambiguity plays a more significant role in folksonomies that include a broad subject domain. Recommenders for Citeulike need be little concerned with ambiguity, while those for Delicious must be wary.

Redundancy can hamper the effort to judge recommendations

	Delicious		Bibsonomy		Citeulike	
	amb.	red.	amb.	red.	amb.	red.
pop_r	4.658	0.025	2.304	0.152	2.421	0.113
pop_u	4.960	0.002	2.885	0.012	2.486	0.016
knn_{ur}	4.625	0.013	2.356	0.014	2.539	0.041
knn_{ut}	4.601	0.014	2.352	0.014	2.503	0.047
$folk$	4.519	0.013	2.629	0.020	2.612	0.037

Table 5: Average ambiguity and redundancy for five strategies on a recommendation set of 10.

as well. Tags synonymous to those in the holdout set are treated as misses even when they serve the user's need. The quality of a recommendation set may also be negatively effected by redundancy when the inclusion of several redundant tags imposes additional effort on the user. We have discovered that our five recommenders produce varying amounts of redundancy.

In sum, we have demonstrated that tag ambiguity and redundancy hinders the evaluation and utility of tag recommenders in folksonomies. Future work will explore ambiguity and redundancy aware recommendation strategies.

7. 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.

8. REFERENCES

[1] A. Almeida, B. Sotomayor, J. Abaitua, and D. López-de Ipiña. folk2onto: Bridging the gap between social tags and ontologies.

[2] V. Batagelj and M. Zaveršnik. Generalized cores. *Arxiv preprint cs/0202039*, 2002.

[3] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117, 1998.

[4] M. Brun, C. Sima, J. Hua, J. Lowey, B. Carroll, E. Suh, and E. Dougherty. Model-based evaluation of clustering validation measures. *Pattern Recognition*, 40(3):807–824, 2007.

[5] E. Durkheim, F. Alcan, and M. Morente. *De la division du travail social*. Presses universitaires de France Paris, 1960.

[6] J. Gemmell, A. Shepitsen, B. Mobasher, and R. Burke. Personalization in Folksonomies Based on Tag Clustering. *Intelligent Techniques for Web Personalization & Recommender Systems*, 2008.

[7] J. Gemmell, A. Shepitsen, B. Mobasher, and R. Burke. Personalizing navigation in folksonomies using hierarchical tag clustering. In *Proceedings of the 10th international conference on Data Warehousing and Knowledge Discovery*. Springer, 2008.

[8] S. Golder and B. Huberman. The Structure of Collaborative Tagging Systems. *Arxiv preprint cs.DL/0508082*, 2005.

[9] A. Hotho, R. Jaschke, C. Schmitz, and G. Stumme. BibSonomy: A Social Bookmark and Publication Sharing System. *Proceedings of the Conceptual Structures Tool Interoperability Workshop at the 14th International Conference on Conceptual Structures, Aalborg, Denmark, July*, 2006.

[10] A. Hotho, R. Jaschke, C. Schmitz, and G. Stumme. Information retrieval in folksonomies: Search and ranking. *Lecture Notes in Computer Science*, 4011:411, 2006.

[11] R. Jaschke, L. Marinho, A. Hotho, L. Schmidt-Thieme, and G. Stumme. Tag recommendations in folksonomies. *Lecture Notes In Computer Science*, 4702:506, 2007.

[12] S. T. Jeff Z. Pan and E. Thomas. Reducing ambiguity in tagging systems with folksonomy search expansion. In *6th European Semantic Web Conference 2009*, 2009.

[13] M. E. I. Kipp and G. D. Campbell. Patterns and inconsistencies in collaborative tagging systems : An examination of tagging practices. *Proceedings of the American Society for Information Science and Technology*, November 2006.

[14] S.-S. Lee and H.-S. Yong. Component based approach to handle synonym and polysemy in folksonomy. In *CIT '07: Proceedings of the 7th IEEE International Conference on Computer and Information Technology*, pages 200–205, Washington, DC, USA, 2007. IEEE Computer Society.

[15] M. Lipczak, R. Angelova, M. Lipczak, E. Milios, P. Pralat, M. Lipczak, J. Blustein, E. Milios, M. Lipczak, M. Lipczak, et al. Tag Recommendation for Folksonomies Oriented towards Individual Users. *ECML PKDD Discovery Challenge*, page 84, 2008.

[16] S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[17] L. Marinho and L. Schmidt-Thieme. Collaborative Tag Recommendations. In *Proceedings of 31st Annual Conference of the Gesellschaft fr Klassifikation (GfKl), Freiburg. Springer*. Springer, 2007.

[18] A. Mathes. 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*, 2004.

[19] P. Mika. 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, 2007.

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

[21] T. Vander Wal. Folksonomy definition and wikipedia. November 2005.

[22] J. Vig, S. Sen, and J. Riedl. Tagsplanations: explaining recommendations using tags. In *IUI '09: Proceedings of the 13th international conference on Intelligent user interfaces*, pages 47–56, New York, NY, USA, 2009. ACM.

[23] X. Wu, L. Zhang, and Y. Yu. Exploring social annotations for the semantic web. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 417–426, New York, NY, USA, 2006. ACM.

[24] C. Yeung, N. Gibbins, and N. Shadbolt. Tag meaning disambiguation through analysis of tripartite structure of folksonomies. In *Proceedings of the 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology-Workshops*, pages 3–6. IEEE Computer Society Washington, DC, USA, 2007.

[25] L. Zhang, X. Wu, and Y. Yu. Emergent semantics from folksonomies: A quantitative study. *Journal on Data Semantics*, pages 168–186, 2006.