Pragmatic Evaluation of Folksonomies

Denis Helic
Graz University of Technology
Graz, Austria
dhelic@tugraz.at

Markus Strohmaier
Graz University of Technology
and Know-Center Graz
Graz, Austria
markus.strohmaier@tugraz.at

Christoph Trattner
Graz University of Technology
Graz, Austria
ctrattner@iicm.edu

Markus Muhr
Know-Center Graz
Graz, Austria
mmuhr@know-center.at

Kristina Lerman
University of Southern
California
Marina del Rey, CA, USA
lerman@isi.edu

ABSTRACT

Recently, a number of algorithms have been proposed to obtain hierarchical structures — so-called folksonomies — from social tagging data. Work on these algorithms is in part driven by a belief that folksonomies are useful for tasks such as: (a) Navigating social tagging systems and (b) Acquiring semantic relationships between tags. While the promises and pitfalls of the latter have been studied to some extent, we know very little about the extent to which folksonomies are *pragmatically useful* for navigating social tagging systems. This paper sets out to address this gap by presenting and applying a pragmatic framework for evaluating folksonomies. We model exploratory navigation of a tagging system as decentralized search on a network of tags. Evaluation is based on the fact that the performance of a decentralized search algorithm depends on the quality of the background knowledge used. The key idea of our approach is to use *hierarchical structures* learned by folksonomy algorithms as *background knowledge* for decentralized search. Utilizing decentralized search on tag networks in combination with different folksonomies as hierarchical background knowledge allows us to evaluate navigational tasks in social tagging systems. Our experiments with four state-of-the-art folksonomy algorithms on five different social tagging datasets reveal that existing folksonomy algorithms exhibit significant, previously undiscovered, differences with regard to their utility for navigation. Our results are relevant for engineers aiming to improve navigability of social tagging systems and for scientists aiming to evaluate different folksonomy algorithms from a pragmatic perspective.

Categories and Subject Descriptors

H.5.3 [**Information Interfaces and Presentation**]: Group and Organization Interfaces—*Collaborative computing*; H.5.4 [**Information Interfaces and Presentation**]: Hypertext/Hypermedia—*Navigation*

General Terms

Algorithms, Experimentation, Measurement

Keywords

folksonomies, evaluation, decentralized search, navigation

WWW 2011, March 28–April 1, 2011, Hyderabad, India.
ACM 978-1-4503-0632-4/11/03.

1. INTRODUCTION

In recent years, social tagging systems have emerged as an alternative to traditional forms of organizing information. Instead of enforcing rigid taxonomies with controlled vocabulary, social tagging systems allow users to freely choose so-called *tags* to annotate resources. In past research, it has been suggested that social tagging systems can be used to acquire latent hierarchical structures that are rooted in the language and dynamics of the underlying user population [7, 3, 13, 14]. The notion of "folksonomies[1]" - from folk-generated taxonomies - emerged to characterize this idea.

A number of algorithms have been proposed in the past to obtain folksonomies from social tagging data [13, 22, 3]. Such folksonomies could potentially be useful for a number of tasks, including: (a) Navigating unstructured information collections, such as social tagging systems and (b) Acquiring semantic relationships between tags. While the promises and pitfalls of the latter have been studied to some extent ([6, 20, 21]), to the best of our knowledge there is no comprehensive attempt to assess the extent to which folksonomies are *pragmatically useful* for tasks such as navigation. This paper sets out to address this gap.

As the main contribution of this paper, we introduce a novel framework for the *pragmatic* (i.e. task-oriented) evaluation of folksonomies. This framework is completely general. It can be used to measure the performance of some folksonomy on some navigation task according to some predefined metric for a given dataset. In this paper, we illustrate the framework by evaluating the performance of four different folksonomy algorithms on an *exploratory* navigation task for five different datasets. Specifically, we view exploratory navigation in a tagging system as a *decentralized search*, an approach originally developed to model and evaluate searchability of *social* [17] and *communication* networks [1]. We show the theoretical suitability of folksonomies for supporting decentralized search, and put them to a navigational task by using them as background knowledge for exploratory navigation. We simulate exploratory navigation behavior (browsing) of users in tagging systems [32] in the following way: In each simulation, an agent's task is to navigate from a starting resource node to a set of resources that are weakly-connected through some common topics (e.g. all resources related to `Toronto, university, campus`). The agent navigates the system with local knowledge (local neighbourhood of the tag graph) and hierarchical background knowledge (a given folksonomy) only. Then, the extent to which an agent can successfully identify short paths between a starting node and the

[1] http://www.vanderwal.net/folksonomy.html

target resources (using local and background knowledge), and the agent's efficiency in doing so is indicative of the pragmatic utility of a given folksonomy for exploratory navigation. The agents use a search strategy based on Kleinberg's decentralized search algorithm with hierarchical background knowledge [19], where the *output* produced by different folksonomy algorithms (i.e. hierarchical structures) is used as an *input* to a decentralized search algorithm (hierarchical background knowledge). Such an approach allows us to answer two important questions related to folksonomies: (a) Can folksonomies inform efficient navigation in social tagging systems? and if so, (b) Do state-of-the-art folksonomy algorithms exhibit differences in their performance on this task?

Our results show that existing folksonomy algorithms differ significantly with regard to their utility for exploratory navigation, which requires new ways of thinking about mechanisms for folksonomy induction and evaluation. Our results suggest that *pragmatic* evaluation represents an important complement to existing *semantic* evaluation strategies for emergent taxonomic structures (such as semantic evaluation [8]).

In our previous work on semantic evaluation of folksonomies we introduced a framework that compares learned folksonomies to a reference hierarchy [26, 28]. We used two metrics - Lexical Recall and Taxonomic Overlap. Lexical Recall computes recall - how many terms exist in both the learned folksonomy and reference directory. Our Taxonomic Overlap is an adapted measure of that measure introduced in [23]. It computes how many parent-child pairs are in correct order. As reference taxonomy we used DMOZ (Open Directory Project).

Further measures for semantic evaluation of conceptual hierarchies include the Augmented Precision & Recall [10] and OntoRand [5]. Augmented Precision & Recall can be divided into a global and a local measure. The measures compare two concepts based on their distance in the hierarchy, i.e. the height of their least common ancestor. Further developments adopt the same approach but take into account e.g. the hierarchy branching factor. On the other hand, OntoRand is a symmetric measure extending hierarchical clustering methods for comparing two partitions of instances. In details, OntoRand has two alternative possibilities to measure similarity of concept hierarchies: the first investigates common ancestors of two concepts, whereas the second one is, similarly to Augmented Precision & Recall, based on the distance (represented through the height of their least common ancestor) between two concepts in the hierarchy. What these approaches have in common is a focus on analyzing semantic aspects, analyzing the pragmatic utility of folksonomies represents a new perspective on folksonomy evaluation.

The paper is structured as follows: First, we will explain Kleinberg's decentralized search, and how it ties into our evaluation approach. After that, we validate the framework by applying it to four folksonomy induction algorithms on five different datasets. Finally, we conclude by discussing implications for folksonomy research.

2. DECENTRALIZED SEARCH

The basic idea of our framework is to use the output produced by different folksonomy algorithms (i.e. hierarchical structures) as input (background knowledge) for decentralized search in social tagging systems. Decentralized search assumes that a search agent only has local knowledge of the network structure, i.e., no knowledge of the network beyond its immediate 1-hop neighbourhood. As such, *decentralized search is a natural model of the user navigation in hypertext* systems where users at any given page are only aware of the links emanating from that page and users usually do not posses any knowledge whatsoever about links from other pages in the system. Therefore, *decentralized search represents a very natural model of navigating* tagging systems.

Decentralized Search. In decentralized search on a network, an algorithm starts its search at an arbitrary *start node* and tries to reach an arbitrary *destination node*. Search is carried out by moving along the links in the network in a number of intermediate steps. At each step, the decision which links to follow is made based on *local knowledge* of the network only. In other words, apart from the destination node, the search algorithm knows only the immediate neighbors.

Research on decentralized search was, for the most part, inspired by Milgrams's "small world experiment" [25]. In this experiment, selected people in Nebraska received a letter they were then asked to send through their social contacts to a stockbroker in Boston. The striking result of the study was that, for those letters reaching the destination, the average number of hops was around 6, i.e. the population of the USA constituted a "small world."

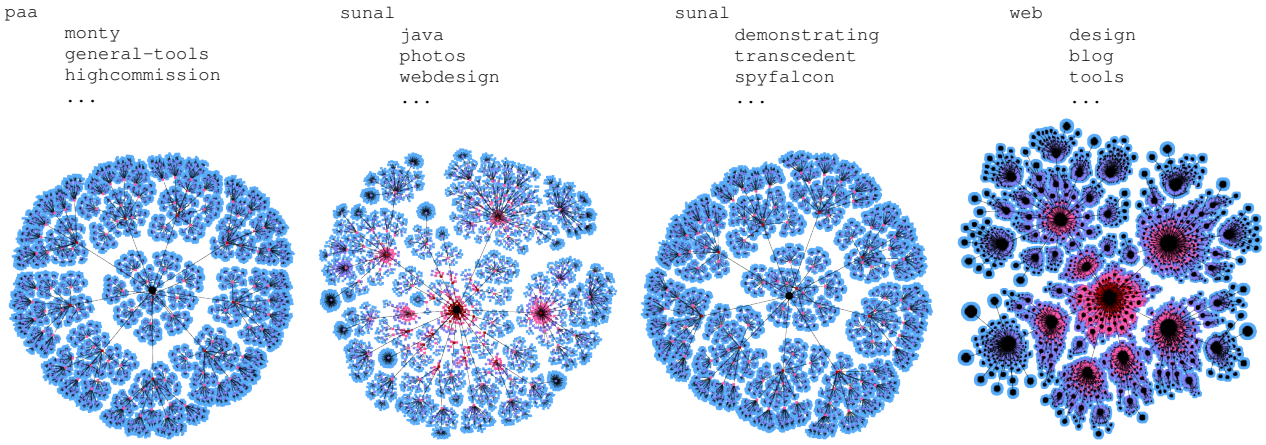
Hierarchical Background Knowledge. Later, Kleinberg analyzed an implicit result of the Milgram's experiment, the ability of humans to *find a short path* when there is such a path between two nodes [18, 16, 19]. Kleinberg concluded that social networks possess certain latent properties that humans are aware of. This *background knowledge* of network structure allows humans to find a short path between two arbitrary network nodes efficiently. Kleinberg defined an "efficiently" searchable network as a network for which a *decentralized search algorithm* exists, such that its delivery time (the number of nodes that the algorithm needs to visit before it reaches the destination node) is polynomial in $logN$, where N is the number of nodes in the network.

Subsequent work has investigated the nature of background knowledge that is required for efficient decentralized search algorithms. In other words: What structural properties do efficiently searchable networks possess? To that end, Kleinberg designed a number of network models such as the 2D-grid model [16], hierarchical model [19], and group model [19]. Independently, Watts [34] introduced the notion of social identity as a membership in a number of social groups organized in hierarchies and showed the existence of efficient decentralized search algorithms by simulation.

Both of these hierarchical network models are based on the idea that, in many settings, the nodes from a network are organized in a taxonomy (Kleinberg's model) or a number of independent taxonomies (Watts' model). The taxonomies can be represented as b-ary trees where network nodes are attached to the leaves of the trees. The basic feature of these models is then the notion of *distance* between two nodes in the network. Kleinberg defines the distance between two nodes v and w to be the height $h(v, w)$ of the least common ancestor of v and w in the tree. Watts defines the distance between two nodes to be the minimum tree distance (in the sense of the height of the least common ancestor of these two nodes) over all model hierarchies.

The crucial structural property of the class of searchable networks is that the probability of two nodes being connected by a link *decreases with their hierarchy distance*. Nodes are highly interlinked locally with other nodes from their immediate hierarchy neighborhood. On the other hand, there are only a few so-called *long-range links* between any given node and more distant nodes (however, such long-range links keep the network connected and are essential for the existence of short paths in the network). This structural property can be formally introduced as a probability linking distribution defined as a function of node distance. Thus, in searchable networks the probability that nodes v and w are connected by a link decreases exponentially with $h(v, w)$.

Next, Kleinberg (theoretically) [19] and Watts (by simulation)



(a) Random (b) Affinity Propagation (c) Hierarchical K-Means (d) DegCen/Cooc

Figure 1: Examples of folksonomies obtained from tagging data using (a) Random (b) Affinity Propagation (c) Hierarchical K-Means and (d) Tag Similarity Graph (DegCen/Cooc) algorithms. The different algorithms produce significantly different folksonomies, their pragmatic usefulness for tasks such as navigation is generally unknown. The visualizations include the top four folksonomy levels of the Delicious dataset. The color gradient starts at red for the top level and proceeds to blue for the fourth level – DegCen/Cooc produces broader hierarchies that other algorithms, and Aff. Prop. hierarchies are broader than K-Means on the first few levels.

[34] showed that for networks with such link probability distributions efficient sub-linear decentralized search algorithms exist. The algorithm starts at an arbitrary start node and moves to an arbitrary destination node by adopting a simple greedy searching strategy. At each time step the algorithm moves to a neighbor node that is closest to the destination node, i.e., it is at the smallest hierarchy distance to it. The basic idea behind such a greedy strategy is that there is a high probability to find a link to the destination node in its immediate neighborhood, simply because local links are abundant in the network.

Utility of Background Knowledge. In [1] Adamic investigated decentralized search in social networks. Adamic conducted a series of experiments by simulating search in an organizational e-mail network and an online student network. The simulations used different hierarchies as background knowledge, e.g., for search in the e-mail network an organizational hierarchy and a hierarchy reflecting the position of a person in the physical space have been applied. Results showed that both of these hierarchies can be effectively used to support decentralized search, but in one case (the online student network), the simulation results were less successful.

3. EVALUATION FRAMEWORK

An important result of Adamic's experiments is the discovery that the *performance of a decentralized search algorithm depends on the quality of the hierarchical background knowledge*. These findings are consistent with Milgram's original "small world" experiment. Travers [33] analyzed the letter chains that reached the target by dividing them into two groups: those that reached the target through the professional contacts and those that reached the target through geography. On average, those that reached the target through geographical assumptions needed more steps. The difference in the number of steps was found to be statistically significant.

Our folksonomy evaluation framework is based on this insight. The performance of an agent's navigation task where the agent uses folksonomies as background knowledge depends on the suitability of that folksonomy to find shortest paths between nodes. An agent might perform better (i.e. its delivery time, or its failure rate in finding the target node is smaller) using one folksonomy instead of another. Thereby, two questions about folksonomies can be answered:

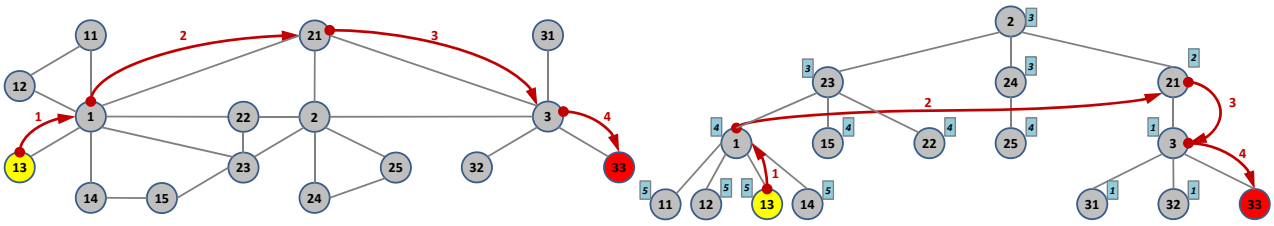
1. Are folksonomies suitable as background knowledge for navigating tagging systems?
2. If a given number of folksonomies are suitable, which one is better?

As navigation can be modeled as decentralized search, the answers to these questions provide insight into *the suitability of folksonomies for navigation from a pragmatic point of view*.

In our framework, a tagging dataset is modeled as a tripartite hypergraph with $V = R \cup U \cup T$, where R is the resource set, U is the user set, and T is the tag set [7, 31, 29]. An annotation of a particular resource with a particular tag produced by a particular user is a hyperedge (r, t, u), connecting three nodes from these three disjoint sets. Such a tripartite hypergraph can be mapped onto three different bipartite graphs connecting users and resources, users and tags, and tags and resources, or onto e.g. tag-tag graphs. For different purposes it is often more practical to analyze one or more of these graphs. For example, in the context of ontology learning, the bipartite graph of users and tags has been shown to be an effective projection [24]. In this paper, we focus on navigating the tag–tag graphs, to mimic tag-based navigation. However, while we limit our investigations to these graphs for practical reasons, our framework supports evaluations of other graphs as well, e.g. bipartite tag-resource graphs.

The pragmatic folksonomy evaluation framework consists of the following steps:

(i) Folksonomy induction. The common objective of folksonomy induction algorithms is to produce a hierarchical structure ("folksonomy") from unstructured data in a tagging system. Such algorithms analyze various evidence such as tag–resource graphs [24], tag-tag graphs [13], tag coocurrence [31], etc to learn hierarchical relations between tags. We describe several folksonomy induction algorithms in greater detail in Sec. 4.2. Examples of folksonomies obtained from a Delicious dataset are shown in Figure 1.



(a) Tag Network (b) Hierarchical Background Knowledge

Figure 2: Decentralized Search: An example of decentralized search in a network of tags (a) using hierarchical background knowledge (b). The tag network links tags if they are used to annotate the same resource. The search begins at the yellow node 13. The destination node is the red node 33. At each step, the search algorithm selects one of the current node's adjacent nodes, which is the closest to the destination node in the hierarchy. The numbers in boxes in (b) provide the distance between the current node and the destination node 33. At step one, node 13 has a single adjacent node 1, so search continues to 1. At step two, 1's adjacent nodes include 11, 12, 13, 14, 15, 21, 22, and 23. The algorithm consults the hierarchy finding out that node 21 is the closest to the destination node. At step three, the algorithm has an option to move to nodes 1, 2, or 3. Search selects node 3 since again, it has the smallest distance to the destination node. Finally, at step 4, the destination node is successfully reached.

(ii) Classification of searchable networks. Next, we calculate a distribution of the distance d between tags in a folksonomy for connected tags in the tag-tag network. This distance distribution is then analyzed to see how it compares with the theoretical class of searchable networks. Watts [34] analyzed a theoretical network model based on an exponential link distribution $ce^{-\alpha d}$ where c is a normalizing constant, α is a tunable parameter, and d is the distance in a socially relevant hierarchy, e.g., a profession hierarchy. Depending on the value of the parameter α, the network might be classified as searchable or unsearchable [34]. The distribution models social networks and the probability of people to be acquainted with other people. The intuition behind the α parameter is that this parameter measures the tendency of people to be acquainted with other "similar" people. When $e^{-\alpha} \ll 1$, the generated network consists of disconnected cliques, i.e., the world is completely homophilous. On the other hand, if any person has the same probability to be acquainted with any other person (yielding a random network), then $e^{-\alpha} = b$ where b is the branching factor of the hierarchy in question. The distance distribution in this case takes the form b^d. Watts showed that almost all searchable networks display $\alpha > 0$ and are situated between these two extremes. The searchable networks are essentially homophilous but not completely so, i.e., there is always a certain number of long-ranged links that connect different cliques with each other. By applying this analysis to folksonomies, we can assess the theoretical suitability of folksonomies for decentralized search.

(iii) Modeling navigation. We select a number of nodes (here: 100,000 resource nodes) uniformly at random from the tagging network. Each of these nodes represents a *starting node* for decentralized search, modeling an arbitrary user entry page into the system (e.g. a landing page from a search engine, the latest resource from a news feed, homepage, or similar). We assume that users who come to the tagging system and do not have their information need satisfied would *explore* the system to find one or more related topics or resources of current interest. To model this, we select another resource node from the tagging network uniformly at random. Tags associated with the second resource are both related to each other (they overlap at least at the second resource) and represent a collection of related resources that a user might be interested in. Therefore, we define those tags as *target nodes* for the search agent. Henceforth, we will call the pair of a start node and a set of target nodes a *search pair*. The goal of the agent is to find a short path from the starting node to one of the target nodes in the search pair.

(iv) Defining evaluation metrics. We use *length of the shortest path* as the performance metric in the evaluation. This reflects a typical scenario of exploratory search. In case that the landing page (start node) does not satisfy a user's information need, the user will explore the tagging system by navigating to related tags in order to find relevant topics and resources *as quickly as possible*, i.e., with as few clicks as possible. We calculate the global shortest path between nodes from each search pair using breadth first search. If there is no global path between nodes from a pair (i.e. when one of the target nodes does not belong to the giant component) then this node is removed from future calculations. The global shortest path between nodes is used later on as a reference value for measuring the effectiveness of decentralized search.

(v) Simulation. We simulate exploratory navigation by performing decentralized search using a greedy search strategy on the search pairs. The folksonomy is applied as background knowledge to provide the notion of distance between nodes. The distance is calculated as proposed in [1]. The parent node and the sibling nodes are considered to be at distance $d = 1$. From there on, the distance is recursively assigned, e.g., the parent's siblings are at distance $d = 2$, the children of the parent's siblings are at distance $d = 3$ and so on. An illustrative example is shown in Figure 2. Although search starts at a resource node, as soon as the first tag is selected, the search becomes a search in the tag–tag network. At each step, the algorithm knows all resources associated with the current tag, as well as all tags of those current resources (this models a typical user interface in a tagging system where a resource is always displayed with tags associated with it). Search is considered successful if the algorithm finds at least one of the target tags. To model users behavior in exploratory navigation, the following strategies are applied by the search agent:

1. If the agent arrives at a certain node for the second time, the search stops and is counted as a failure (no backtracking) – this mimics the situation where a user arrives at a tag that he already visited, and then decides to, e.g., switch to the search field or to leave the system.
2. In the case of a distance tie (two or more tags are equally close to a target node) the highest degree tag is selected as the next hop – in tagging systems tags are typically sorted by degree and this models a user selecting the first tag from the sorted list.

3. If the agent did not find a target node in at least n steps (a tunable parameter), then the search stops and it is again counted as a failure – this models users who loose the motivation to continue exploring the system.

The success rate thereby provides an answer to the question of the pragmatic suitability of a folksonomy to support navigation.

(vi) Evaluation. Finally, we compare the results of simulation with the defined evaluation metrics (here: the global shortest path) and the difference in the number of hops needed by the simulator is calculated for each of the simulated pairs. In the final step, the simulation results for different folksonomies are compared to each other. In addition to these steps, a number of adaptations can easily be accommodated by the framework. For example: in step (iv), different evaluation metrics can be selected or in step (v), real world data can be used instead of simulations. We will briefly discuss these adaptations next:

Alternative evaluation metrics. While the global shortest path is a useful metric to evaluate how a folksonomy supports exploratory navigation, an *alternative metric* might be adopted to evaluate the usefulness of a folksonomy for an *alternative task*. For example, for the task of finding relevant resources, one could use the number or diversity of resources found instead of the global shortest path metric. This also means that our evaluation would yield different results for different tasks and corresponding metrics. We consider this to be a desirable property of a *pragmatic* evaluation framework.

Simulation vs. Real-World Data. While decentralized search with local knowledge represents an intuitive model of user navigation in networks, the evaluation framework does not depend on the simulation to accurately reflect users' actual navigation behavior in social tagging systems: Instead of simulating exploratory navigation, the evaluation framework could equally use *actual* navigation data (e.g. click trails through a system) from *real* users. In this case, our approach would evaluate which folksonomy best explains given user behavior, and thereby reveal which folksonomy or set of folksonomies is most likely to be suitable for a given user population. This would also mean that evaluation would yield different results for different observed or assumed user behavior. Again, this can be considered a desirable property of a *pragmatic* evaluation framework.

4. VALIDATION

While the framework supports evaluation based on both simulations and actual user data, in this paper we use simulation for better experimental control, better illustration of our framework and due to the difficulty of obtaining actual navigation data for all of our datasets. For validation, we apply the framework to evaluate the pragmatic utility of four different folksonomy induction algorithms on five different social tagging data.

4.1 Datasets

The following datasets were used as an empirical basis:

Dataset BibSonomy: This dataset[2] contains nearly all 916,495 annotations and 235,340 resources (scientific articles) from a dump of BibSonomy [15] until 2009-01-01. The tag-tag network comprises 56,424 tags and 2,003,986 links.

Dataset CiteULike: This dataset[3] contains 6,328,021 annotations and 1,697,365 resources (scientific articles). The tag-tag network comprises 347,835 tags and 27,536,381 links.

Dataset Delicious: This dataset is an excerpt from the PINTS experimental dataset[4]. We extracted all data (resources are URLs) from 11/2006. The tag-tag network consists of 380,979 tags and 39,808,439 links.

Dataset Flickr: This dataset is also an excerpt from the PINTS dataset. It contains the data (resources are photos) from 12/2005. The tag-tag network consists of 395,329 tags and 17,524,927 links.

Dataset LastFm: This dataset is from [30]. It contains annotations from the first half of 2009. The resources in this dataset are songs, artists and albums. The tag-tag network consists of 281,818 tags and 84,787,780 links.

4.2 Folksonomy Algorithms

On these five datasets, we apply and evaluate four state-of-the-art folksonomy induction algorithms. The common objective of these algorithms is to produce hierarchical structures ("folksonomies") from unstructured tagging data. While further algorithms exist (such as [22]), we have selected the following four algorithms because (i) they were well documented and (ii) for their ease of implementation. The evaluation framework can be used to evaluate any kind of folksonomy induction algorithm that produces hierarchical structures as an output. The initial set of four algorithms acts as a demonstration of the evaluation framework's capabilities only. In the following, we briefly describe each algorithm and how it has been applied by us in this paper.

Affinity Propagation (AP) Frey and Dueck introduced Affinity Propagation as a new clustering method in [11]. As input, Affinity Propagation accepts a set of similarities between data samples provided in a matrix. The diagonal entries (self-similarities) of the similarity matrix are called preferences and are set according to the suitability of the corresponding data sample to serve as a cluster center (exemplar called in [11]). Although no explicit cluster number must be set, the preference values correlate with the number of resulting clusters (lower preference values results in fewer clusters and vice versa). AP runs by exchanging messages between data samples to update their "responsibility" and "availability" values. Responsibility values reflect how well data samples serve as exemplars for other data, and the availability values show the suitability of other data samples to be the exemplars for specific data samples. Responsibility and availability are refined iteratively with a parameter λ as an update factor.

In previous work [27], we have introduced an adaption of affinity propagation to infer a taxonomy. We incorporated structural constraints directly into the global objective function of affinity propagation, so that a tree evolves naturally from execution. In this paper, we follow a simpler approach by applying the original AP recursively in a bottom-up manner. In a first step, the top 10 Cosine similarities (pruned for memory reasons) between the tags in a given data set serve as the input matrix, and the minimum of those serves as preference for all data samples. Then, AP produces clusters by selecting examples with associated data samples. If the ratio between number of clusters and data samples is between 3 and 15 (adjustable parameter), then the result will be retained, otherwise another run with lower (too many clusters have been selected) or higher preference values (too few clusters have been selected) will be executed. Then, the centroids of the clusters are calculated by using the sum of the connected data samples normalized to unit length. Now the Cosine similarities between the centroids serve as input matrix for the next run of affinity propagation. This approach is executed until the top-level is reached. Since we want a tag hi-

[2]http://www.kde.cs.uni-kassel.de/ws/dc09/

[3]http://www.citeulike.org/faq/data.adp

[4]https://www.uni-koblenz.de/FB4/Institutes/IFI/AGStaab/Research/DataSets/PINTSExperimentsDataSets/

erarchy where each node represents a unique tag, a sample in each cluster is used as describing tag. The tag representing a node is selected by taking the nearest tag to the centroid. Furthermore, this tag is removed from the actual tags contained in the leaf cluster and is not used as representative in lower hierarchy levels. As parameter settings, we set λ_0 to 0.6 with increasing values depending on the iteration count (i) ($\lambda_i = \lambda_{i-1} + (1.0 - \lambda_0) * i/i_{max}$). AP will terminate after either a maximum of 5000 iterations (i_{max}) or if the exemplars of clusters are stable for at least 10 iterations.

Hierarchical K-Means Dhillon et al [9] introduced an adaption to the k-means algorithm for textual data by optimizing the Cosine similarity instead of Euclidean distance [9], while [35] introduced an efficient version of an online spherical k-means. Without going into detail, these adaptations allow an online version to be at least as fast as a batch spherical k-means with better results. We utilize k-means iteratively in a top-down manner to build a tag hierarchy. Basically, in the first step, the whole input data set is used for clustering the data into 10 clusters. Clusters containing more than 10 connected samples are further partitioned while ones with less than 10 samples are considered as leaf clusters. However, since a cluster set of 11 samples would also be partitioned into 10 clusters we introduced a special case to give some freedom to the clustering process for these border cases by setting the cluster number to the maximum of 10 or number of data samples divided by 3 what would result in 3 clusters in case of 11 samples. The tag representing a node is selected by taking the nearest tag to the centroid. Furthermore, this tag is removed from the actual tags contained in a cluster and which are further clustered in the next step, if there are more than 10 samples left.

Generality in Tag Similarity Graph (Closeness Centrality / Cosine Similarity) In [13], the authors describe an algorithm developed to overcome the limited success in producing hierarchical structures from the tagging data by means of hierarchical clustering. The input for the algorithm is the so-called *tag similarity graph* – an unweighted graph where each tag is a node in the graph, and two nodes are linked to each other if their similarity is above a predefined similarity threshold. In the simplest case, the threshold is defined through tag overlap – tags need to share at least one resource to be linked in the tag similarity graph. The second prerequisite for the algorithm is the ranking of nodes in a descending order according to how central the tags are in the tag similarity graph. In particular, this ranking produces a generality order where the most general tags from a dataset are in the top positions. The algorithm starts by a single node tree with the most general tag as the root node. The algorithm then proceeds by iterating through the generality list and adding each tag to the tree – the algorithm calculates the similarities between the current tag and each tag currently present in the tree and adds the current tag as a child to its most similar tag. The authors describe their algorithm as extensible as they leave the possibility to apply different similarity, as well as different centrality measures. The presented algorithm works with cosine similarity and closeness centrality, and we denote this algorithm henceforth CloCen/Cos.

Generality in Tag Similarity Graph (Degree Centrality / Co-Occurrence) In [3], the authors describe an extension of the algorithm presented in [13]. Generally, this new algorithm is based on principles similar to Heymann's algorithm – but the new algorithm applies tag co-occurrence as the similarity measure and the degree centrality as the generality measure (DegCen/Cooc). In particular, the algorithm executes an extensive preprocessing of the dataset e.g. to remove synonym tags or to resolve ambiguous tags. For reasons of simplicity, we skipped preprocessing of the dataset and only applied the alternative similarity and centrality measures.

4.3 Folksonomy Evaluation

We evaluate the folksonomies produced by the four different algorithms both on a theoretical and on a pragmatic level.

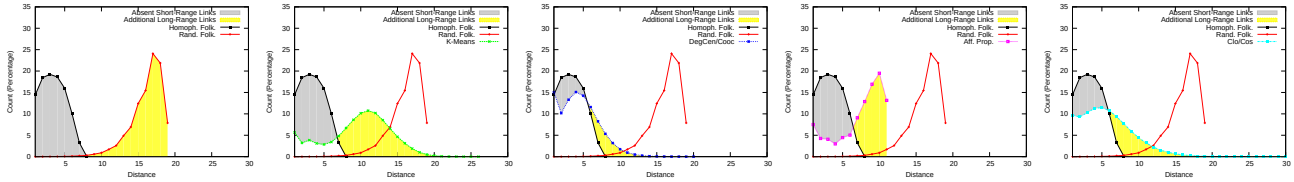
4.3.1 Theoretical suitability of folksonomies

For each pair of connected nodes in the tag-tag network, we measure the distance between the same pair of nodes in a given folksonomy. Analyzing the resulting distribution of distances provides insights into the theoretical suitability of a given folksonomy to support decentralized search. Intuitively, a distance distribution that is dominated by short range distances with occasional long range links represents suitable background knowledge for decentralized search. Specifically, we compare the distance distributions of different folksonomies to the class of theoretically searchable networks determined by a specific range of the α parameter in the exponential distribution $ce^{-\alpha d}$ by Watts. However, we cannot directly compare these distributions without adapting the Watts' model to the specifics of tagging networks. From [12] and related work, we know that the tag degree distribution in a tagging system is a power-law distribution, whereas in Watts' model the degree distribution is uniform. Another difference is that in a folksonomy, tags are potentially attached everywhere in a hierarchy, whereas in Watts' model they would only be attached to leaves.

So in order to adapt Watts' model to tagging networks, we discuss the distance distributions of two synthetic folksonomies that represent a random and a "homophily" scenario. While the "homophily" distance distribution (the distance distribution of isolated cliques) mimics a folksonomy that only supports short range links in the tag-tag network, the random distance distribution mimics a folksonomy that has random (short and long range) links. This is illustrated in Figure 3 where (a) shows the two synthetic distance distributions (Homoph. & random). Neither of these two synthetic folksonomies are optimal: while the distance distribution of the homoph. folksonomy is dominated by short range links, the random folksonomy is dominated by long range links. To be useful as background knowledge for decentralized search, folksonomies need to mostly short range links mixed with occasional long range links.

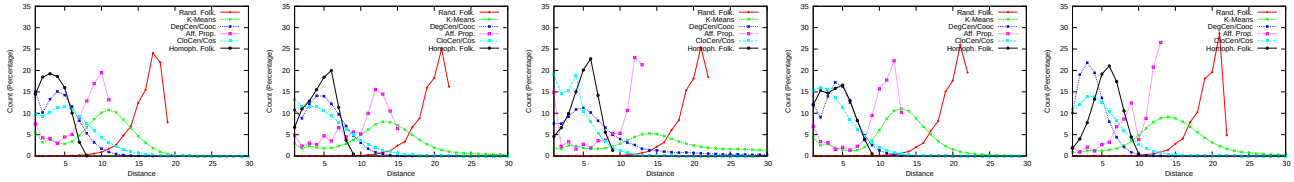
In a random network, any node is equally likely to be linked to any other node, which results in the distance distribution to fall within the range $[b^d, 2 * b^d]$. As, according to Watts [34] and Kleinberg [19] searchable networks exhibit $\alpha > 0$ and since e.g. $e^{-\alpha} = b \implies \alpha < 0$, a random network is not efficiently searchable. Therefore, any folksonomy yielding a distance distribution close to this range renders the network less searchable. In general, as tag-tag networks are power law networks sub-linear decentralized search strategies exist for such networks. For example, Adamic designed a decentralized search algorithm that utilizes the node degree to find a specific target node in the network [2]. The algorithm adopts a simple greedy strategy by moving to an adjacent node of the highest degree. Thus, the algorithm is able to move quickly to a network hub that, with a high probability, has a link to the target node. Although such an algorithm makes a random power law network theoretically searchable [19], within the scope of our framework we consider such a network to be less practical due to a lack of semantic clues. In particular, as our framework models exploratory navigation, utilizing high degree nodes would involve users in exploring thousands of links emanating from a network hub – a task that is practically not feasible.

On the other hand, in a homophilous network of isolated cliques, a node is connected to nodes that are at distance $d = 1$. How-



(a) Homoph. & Rand. (b) K-Means (c) DegCen/Cooc (d) Aff. Prop. (e) CloCen/Cos

Figure 3: Comparison of distance distributions for four folksonomy induction algorithms on the BibSonomy dataset and two synthetic folksonomies: Homoph. Net. (black curve) and Rand. (red curve). Useful distance distributions trade some short range links against long range links to improve the searchability of the network. They are thereby much more similar to the distance distributions of Homoph. folksonomies than to Rand. folksonomies.



(a) BibSonomy (b) CiteULike (c) Delicious (d) Flickr (e) LastFm

Figure 4: Comparison of distance distributions for four folksonomy induction algorithms on five different datasets. Across all datasets, Aff.Prop. and K-Means exhibit distance distributions that are less efficient, i.e. more similar to a random network (dominated by too many long range links). At the same time, DegCent/Cooc and CloCent/Cos distributions are dominated by short range links combined with a few long range links, which renders them more useful for decentralized search.

ever, in a power-law network the high degree nodes typically have $degree \gg 2 * b$ and are therefore also connected to nodes that are at longer distances. To mimic the homophily case, such a high degree node is then firstly connected to all available nodes at distance $d = 1$, then to nodes at distance $d = 2$, then to nodes at distance $d = 3$, and so on until all of its links are assigned. Theoretically, a suitable folksonomy possesses a distance distribution which approximates the homophilous folksonomy, but trades some short range links for long range links. In this sense, the distance distribution of suitable folksonomies are closer to the homophilous folksonomy than to the random one.

To assess the theoretical suitability of different folksonomies for decentralized search we plot the distance distribution first. We then compare the resulting distance distributions with the synthetic distance distributions discussed above. Figure 3 shows a comparison of the "homophily" distance distribution with branching factor $b = 3$ and distance distributions calculated for the folksonomies learned from the Bibsonomy dataset. The grey areas represent the difference in the number of short-range links between the clique and a particular distribution whereas the yellow areas represent the difference in the number of long-range links. As we can expect fewer short-range links in the case of searchable folksonomies, we call this area Absent Short-Range Links area. By analogy, we call the area where additional long range links are introduced the Additional Long-Range Links area. Theoretically, both of these areas need to be greater than 0 but still rather small, i.e., if they are too large the distance distribution is composed of too many long-range links and becomes similar to the random distance distribution (the red curve in Figure 3), which is suboptimal. From Figures 3c and 3e we can observe that DegCen/Cooc and CloCen/Cos distributions exhibit the desired properties (many short range links mixed with a few long range links) on the Bibsonomy dataset, which renders them more suitable than K-Means or Aff. Prop. (Figures 3b and 3d). Figure 4 presents the results for all five datasets.

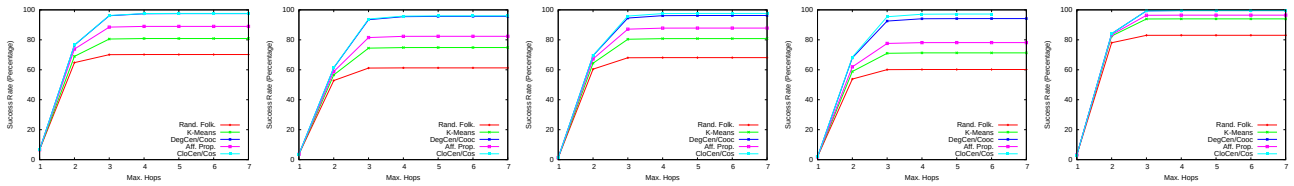
Summary: Existing folksonomy algorithms produce folksonomies that are theoretically useful to support decentralized search. Not all folksonomies are equally useful. Folksonomies produced by tag similarity graph algorithms (DegCen/Cooc and CloCen/Cos) are theoretically more useful than folksonomies produced by hierarchical clustering algorithms (K-Means and Aff. Prop.)

4.3.2 *Pragmatic suitability of folksonomies*

Watts has identified a broad parameter space that is occupied by searchable networks [34]. In other words, analyzing the theoretical suitability of folksonomies for decentralized search only provides a general answer to the question whether a folksonomy falls into this broad region or not. Although the theoretical analysis provides some insights, a *pragmatic* evaluation of folksonomies can not be answered *theoretically*. The answer depends on additional factors such as the task or properties of the tagging network including e.g. degree distribution, the size of the giant component or the shortest path distribution. Therefore, pragmatic analysis is needed.

In the following, we will evaluate the usefulness of folksonomies to support exploratory navigation in tagging systems by simulation. We model exploratory navigation as a process where an agent navigates from a starting resource node to a set of resources that are weakly-connected through some common topics. We study the success rate, i.e. the number of times an agent is successful in finding a path between those nodes, using different folksonomies as background knowledge. Figure 5 presents the success rate of exploratory navigation as the function of a tunable parameter n, the maximal number of steps an agent is allowed to perform before stopping (e.g., an agent only follows n links). All four folksonomies have much better success rates than the random folksonomy (Note that an agent can only be successful if the shortest path between the source and the target node is shorter than n).

While the success rate provides interesting information, we don't



(a) BibSonomy (b) CiteULike (c) Delicious (d) Flickr (e) LastFM

Figure 5: Success rate as the function of the maximal number of steps (hops) n. The success rate of exploratory navigation with a random baseline folksonomy as hierarchical background knowledge is high (60% and higher). Across all datasets and different folksonomies, 70% of the resources can be reached from an arbitrary starting node with 3 hops. With the same number of hops, DegCen/Cooc and CloCen/Cos have success rates of $> 90\%$ or higher on all datasets. They significantly outperform K-Means and Aff.Prop. consistently, sometimes by large margins (d).

know how efficient the agent is, i.e. how often an agent does not find the global shortest path, but some other path that is longer. For that purpose, Figure 6 plots the difference between shortest paths in the system (using global knowledge) and the paths that agents have found (using local knowledge only). The light-blue bars in the histograms are those search pairs where the agent finds the shortest possible path, the green bars are those search pairs where the agent can only find a path that is one hop longer. The dark blue bars refer to paths where it takes the agent two extra hops, and the violet bars refer those cases with three or more extra hops.

As a baseline, we perform exploratory navigation with a randomly generated folksonomy (with branching factor $b = 3$) to obtain a lower bound, depicted in Figures 6a, 6f, 6k, 6p, and 6u. The main cause why an agent using a random folksonomy as background knowledge is considerably successful is the fact that tagging networks are highly connected and have a low effective diameter (< 3.5) [12]. Due to high link density, the majority of tags are connected by *multiple* short paths. That means that even if the agent takes a single non-optimal or wrong link towards the destination tag, with high probability there exists an alternative link which also leads to the destination tag. In particular for the (global) shortest path of 2, an agent using a random folksonomy is considerably successful in finding short path – regardless of the first tag selected, that tag is in the majority of cases linked to the destination tag. However, as the path towards the destination becomes longer (≥ 3) the ability of an agent using a random folksonomy as background knowledge deteriorates. The random folksonomy applied on the LastFM dataset exhibits the most extreme behavior in this respect – since tags in this dataset are music genres their overlap is extremely high. However, across all datasets we see that agents using folksonomies produced by the introduced algorithms find significantly shorter paths than when using a random folksonomy.

Summary: Existing algorithms produce folksonomies that are more useful for exploratory navigation than a random baseline folksonomy. Folksonomies obtained by tag similarity graph methods perform better in supporting exploratory navigation than folksonomies obtained by hierarchical clustering methods. This pragmatic result supports the theoretical results presented in the previous section.

4.3.3 Discussion

Structurally, K-Means hierarchies are typically unbalanced. We performed additional experiments and introduced a balancing factor to resolve these structural issues and obtain more balanced clusters. Preliminary results show that this approach improves the success rate of decentralized search only marginally (e.g. the success rate improves by 1% with BibSonomy dataset), and thereby does not seem to have a significant impact on the validity of our results.

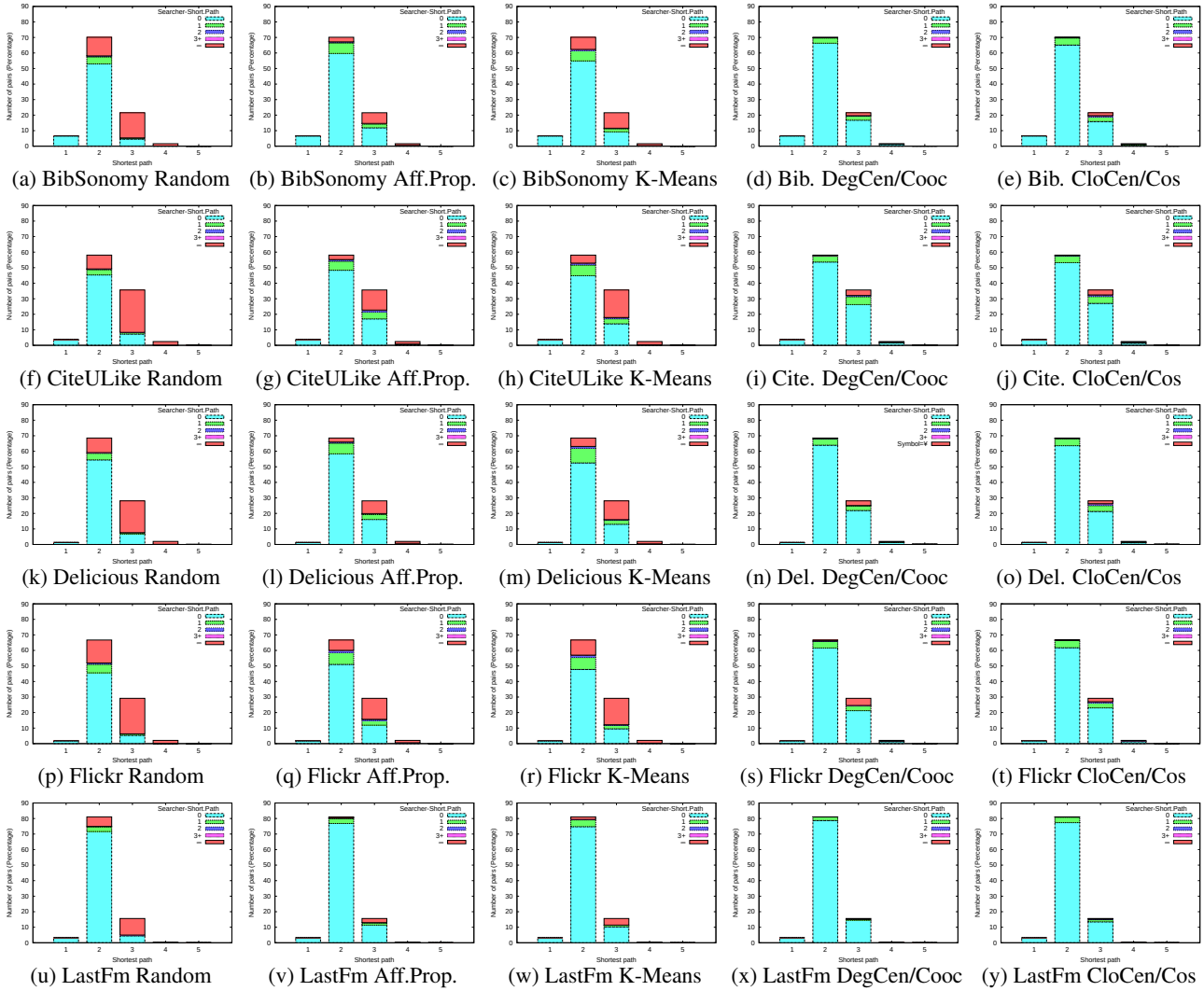
A problem with both Aff. Prop. and K-Means seems to be the choice of the cluster representative. In the current implementation, the cluster representative is chosen by taking the nearest sample to the centroid. As the similarities in tagging datasets are often small and sparse, the similarities between cluster members are equal, and thus the selection of the cluster representative, and thereby a parent node for that cluster in the resulting hierarchy, is completely arbitrary – this could be the main cause why Aff.Prop. and K-Means are inferior to tag graph similarity algorithms. The same issues seem to influence the construction of the Aff.Prop. hierarchy that is based on the similarity between the centroids of the previous execution steps. One possible remedy for this could be to use an average similarity of connected data samples. An advantage of Aff. Prop. over K-Means is that on the upper hierarchical levels the algorithm produces broader structures than K-Means. This seems to make them slightly more suitable for exploratory navigation.

Summarizing, hierarchical clustering methods seem to lack additional information about the dataset as given by the tag similarity graph and centrality ranking. Note that while Heymann et al. in [13] came to a similar conclusion based on intuition, our paper provides both a theoretical and an empirical justification for this.

There are no significant differences in performance of DegCen/Cooc and CloCen/Cos combinations. We performed additional experiments and produced folksonomies by combining betweenness centrality and co-occurrence as well as closeness centrality and co-occurrence. The choice of centrality or similarity measure does not significantly influence performance. Any combination of these two measures performs similar. However, calculating closeness or betweenness centrality involves solving of the all-pairs shortest path problem which is a time costly operation. Even fast approximative algorithms for large networks [4] or incremental approximative algorithms (e.g. when a user adds a new tag) are, for an order of magnitude, slower than degree centrality algorithms. Because of term weights recalculation an incremental computation of the cosine similarity matrix requires more time than an incremental computation of the co-occurrence similarity matrix. Thus, for fast folksonomy computation we suggest DegCen/Cooc combination.

5. CONCLUSIONS

We have presented a pragmatic evaluation framework for folksonomies that connects two previously unconnected fields of research, i.e. research on folksonomy algorithms with decentralized search in networks. Our evaluation framework is completely general with regard to the task, data and evaluation metrics adopted. We have demonstrated the viability of this framework by instanti-



(a) BibSonomy Random (b) BibSonomy Aff.Prop. (c) BibSonomy K-Means (d) Bib. DegCen/Cooc (e) Bib. CloCen/Cos

(f) CiteULike Random (g) CiteULike Aff.Prop. (h) CiteULike K-Means (i) Cite. DegCen/Cooc (j) Cite. CloCen/Cos

(k) Delicious Random (l) Delicious Aff.Prop. (m) Delicious K-Means (n) Del. DegCen/Cooc (o) Del. CloCen/Cos

(p) Flickr Random (q) Flickr Aff.Prop. (r) Flickr K-Means (s) Flickr DegCen/Cooc (t) Flickr CloCen/Cos

(u) LastFm Random (v) LastFm Aff.Prop. (w) LastFm K-Means (x) LastFm DegCen/Cooc (y) LastFm CloCen/Cos

Figure 6: Comparison of global shortest paths and the delivery time (number of hops with local knowledge) with different folksonomies. Agents using the random baseline folksonomy (left column) find short paths, but using anyone of the introduced folksonomy algorithms instead (column 2-5) improves delivery time significantly. Again, DegCen/Cooc and CloCen/Cos consistently outperform Aff.Prop. and K-Means across all datasets ("larger light-blue bars").

ating it to evaluate an exploratory navigation task for four different folksonomy algorithms on five social tagging datasets. In our experiments, we find that folksonomies represent suitable background knowledge for exploratory navigation. Our results show that DegCen/Cooc and CloCen/Cos folksonomy algorithms outperform traditional hierarchical clustering techniques on this task.

The results of this paper suggest that in addition to semantic evaluation, future folksonomy research needs to consider pragmatic evaluations as well, in order to examine the *usefulness* of folksonomies for different tasks. While we have evaluated folksonomies in this paper, our framework can be applied to evaluate manually constructed and/or expert taxonomies as well. Although our results make a theoretical and a pragmatic case for folksonomies to be used in user interfaces of tagging systems, the extent to which folksonomies will be successfully used for this purpose depends on other factors as well, such as cognitive, psychological or user interface constraints.

6. REFERENCES

[1] L. Adamic and E. Adar. How to search a social network. *Social Networks*, 27(3):187 – 203, 2005.

[2] L. A. Adamic, R. M. Lukose, A. R. Puniyani, and B. A. Huberman. Search in power-law networks. *Physical Review E*, 64(4):046135 1–8, Sep 2001.

[3] D. Benz, A. Hotho, and G. Stumme. Semantics made by you and me: Self-emerging ontologies can capture the diversity of shared knowledge. In *Proc. of the 2nd Web Science Conference (WebSci10)*, Raleigh, NC, USA, 2010. Web Science Trust.

[4] U. Brandes and C. Pich. Centrality estimation in large networks. *International Journal of Bifurcation and Chaos*, 17:2303–2318, 2007.

[5] J. Brank, D. Madenic, and M. Groblenik. Gold standard based ontology evaluation using instance assignment. In *Proceedings of the 4th Workshop on Evaluating Ontologies*

for the Web (EON2006), Edinburgh, Scotland, May 2006. CEUR-WS.

[6] C. Cattuto, D. Benz, A. Hotho, and G. Stumme. Semantic grounding of tag relatedness in social bookmarking systems. In *Proc. of International Semantic Web Conference 2008*, volume 5318 of *LNAI*, pages 615–631, Heidelberg, 2008. Springer.

[7] C. Cattuto, C. Schmitz, A. Baldassarri, V. D. P. Servedio, V. Loreto, A. Hotho, M. Grahl, and G. Stumme. Network properties of folksonomies. *AI Commun.*, 20(4):245–262, 2007.

[8] K. Dellschaft and S. Staab. On how to perform a gold standard based evaluation of ontology learning. In *Proc. of International Semantic Web Conference 2006*, Athens, GA, USA, November 2006. Springer, LNCS.

[9] I. Dhillon, J. Fan, and Y. Guan. Efficient clustering of very large document collections. In R. Grossman, C. Kamath, and R. Naburu, editors, *Data Mining for Scientific and Engineering Applications*. Kluwer Academic Publishers, Heidelberg, 2001.

[10] Y. L. Diana Maynard, Wim Peters. Metrics for evaluation of ontology-based information extraction. In *Proceedings of the 4th Workshop on Evaluating Ontologies for the Web (EON2006)*, Edinburgh, Scotland, 2006. CEUR-WS.

[11] B. J. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972–976, January 2007.

[12] D. Helic, C. Trattner, M. Strohmaier, and K. Andrews. On the navigability of social tagging systems. In *Proc. of 2010 IEEE International Conference on Social Computing*, pages 161–168, Los Alamitos, CA, USA, 2010. IEEE Computer Society.

[13] P. Heymann and H. Garcia-Molina. Collaborative creation of communal hierarchical taxonomies in social tagging systems. Technical Report 2006-10, Stanford InfoLab, April 2006.

[14] A. Hotho, R. Jaeschke, C. Schmitz, and G. Stumme. Folkrank: A ranking algorithm for folksonomies. In *Proc. FGIR 2006*, pages 111–114, Bonn, Germany, 2006. Gesellschaft Für Informatik.

[15] A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme. Bibsonomy: A social bookmark and publication sharing system. In A. de Moor, S. Polovina, and H. Delugach, editors, *Proceedings of the Conceptual Structures Tool Interoperability Workshop at the 14th International Conference on Conceptual Structures*, pages 87–102, Aalborg, Denmark, July 2006. Aalborg University Press.

[16] J. Kleinberg. The small-world phenomenon: an algorithm perspective. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, STOC '00, pages 163–170, New York, NY, USA, 2000. ACM.

[17] J. Kleinberg. Complex networks and decentralized search algorithms. In *International Congress of Mathematicians (ICM)*, pages 1019–1044, Zürich, Switzerland, 2006. European Mathematical Society Publishing House.

[18] J. M. Kleinberg. Navigation in a small world. *Nature*, 406(6798):845, August 2000.

[19] J. M. Kleinberg. Small-world phenomena and the dynamics of information. In *Advances in Neural Information Processing Systems (NIPS) 14*, page 2001, Cambridge, MA, USA, 2001. MIT Press.

[20] C. Koerner, D. Benz, M. Strohmaier, A. Hotho, and G. Stumme. Stop thinking, start tagging - tag semantics emerge from collaborative verbosity. In *Proc. of the 19th International World Wide Web Conference (WWW 2010)*, Raleigh, NC, USA, Apr. 2010. ACM.

[21] C. Koerner, R. Kern, H. P. Grahsl, and M. Strohmaier. Of categorizers and describers: An evaluation of quantitative measures for tagging motivation. In *21st ACM SIGWEB Conference on Hypertext and Hypermedia (HT 2010), Toronto, Canada, ACM*, New York, NY, USA, June 2010. ACM.

[22] R. Li, S. Bao, Y. Yu, B. Fei, and Z. Su. Towards effective browsing of large scale social annotations. In *Proc. of the 16th international conference on World Wide Web, WWW '07*, page 952, New York, NY, USA, 2007. ACM.

[23] A. Maedche and S. Staab. Measuring similarity between ontologies. In *Proc. Of the European Conference on Knowledge Acquisition and Management - EKAW-2002. Madrid, Spain, October 1-4, 2002*, volume 2473 of *LNCS/LNAI*, Heidelberg, 2002. Springer.

[24] 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.

[25] S. Milgram. The small world problem. *Psychology Today*, 1:60–67, 1967.

[26] A. Plangprasopchok and K. Lerman. Constructing folksonomies from user-specified relations on flickr. In *Proc. of 18th International World Wide Web Conference, WWW '09*, New York, NY, USA, May 2009. ACM.

[27] A. Plangprasopchok, K. Lerman, and L. Getoor. From saplings to a tree: Integrating structured metadata via relational affinity propagation. In *Proceedings of the AAAI workshop on Statistical Relational AI*, Menlo Park, CA, USA, July 2010. AAAI.

[28] A. Plangprasopchok, K. Lerman, and L. Getoor. Growing a tree in the forest: Constructing folksonomies by integrating structured metadata. In *Proc. of the International Conference on Knowledge Discovery and Data Mining (KDD)*, New York, NY, USA, July 2010. ACM.

[29] M. Ramezani, J. Sandvig, T. Schimoler, J. Gemmell, B. Mobasher, and R. Burke. Evaluating the impact of attacks in collaborative tagging environments. In *Computational Science and Engineering, 2009. CSE '09. International Conference on*, volume 4, pages 136 –143, Los Alamitos, CA, USA, aug. 2009. IEEE Computer Society.

[30] R. Schifanella, A. Barrat, C. Cattuto, B. Markines, and F. Menczer. Folks in folksonomies: social link prediction from shared metadata. In *Proc. of the third ACM int. conference on Web Search and Data Mining, WSDM '10*, pages 271–280, New York, NY, USA, 2010. ACM.

[31] C. Schmitz, A. Hotho, R. Jöschke, and G. Stumme. Mining association rules in folksonomies. In *Data Science and Classification: Proc. of the 10th IFCS Conf., Studies in Classification, Data Analysis, and Knowledge Organization*, pages 261–270, Berlin, Heidelberg, 2006. Springer.

[32] J. Sinclair and M. Cardew-Hall. The folksonomy tag cloud: when is it useful? *Journal of Information Science*, 34:15, 2008.

[33] J. Travers and S. Milgram. An experimental study of the small world problem. *Sociometry*, 32:425–443, 1969.

[34] D. J. Watts, P. S. Dodds, and M. E. J. Newman. Identity and search in social networks. *Science*, 296:1302–1305, 2002.

[35] S. Zhong. Efficient online spherical k-means clustering. *IJCNN*, 5:3180–3185 vol. 5, July-4 Aug. 2005.