New Review of Hypermedia and Multimedia
Vol. 00, No. 00, October 2010, 1–31

RESEARCH ARTICLE

Leveraging Search and Content Exploration by Exploiting Context in Folksonomy Systems

Fabian Abel[a*], Matteo Baldoni[b], Cristina Baroglio[b], Nicola Henze[a], Ricardo Kawase[a], Daniel Krause[a], and Viviana Patti[b]

[a]*IVS – Semantic Web Group & L3S Research Center, Leibniz University Hannover, Germany*; [b]*Dipartimento di Informatica, Università degli Studi di Torino, Italy*

(*January 2010*)

With the advent of Web 2.0 tagging became a popular feature in social media systems. People tag diverse kinds of content, e.g. products at Amazon, music at Last.fm, images at Flickr, etc. In the last years several researchers analyzed the impact of tags on information retrieval. Most works focussed on tags only and ignored context information. In this article we present context-aware approaches for learning semantics and improve personalized information retrieval in tagging systems.

We investigate how explorative search, initialized by clicking on tags, can be enhanced with automatically produced context information so that search results better fit to the actual information needs of the users. We introduce the SocialHITS algorithm and present an experiment where we compare different algorithms for ranking users, tags, and resources in a contextualized way.

We showcase our approaches in the domain of images and present the TagMe! system that enables users to explore and tag Flickr pictures. In TagMe! we further demonstrate how advanced context information can easily be generated: TagMe! allows users to attach tag assignments to a specific area within an image and to categorize tag assignments. In our corresponding evaluation we show that those additional facets of tag assignments gain valuable semantics, which can be applied to improve existing search and ranking algorithms significantly.

Keywords: social media; search and ranking; folksonomies; context; personalization; learning semantics

1. Introduction

During the last decade, the tagging paradigm attracted much attention in the Web community. More and more Web systems allow their users to annotate content with freely chosen keywords (*tags*). The tagging feature helps users to organize content for future retrieval (Marlow *et al.* 2006b). Resource sharing systems like Delicious[1], Flickr[2], or Last.fm[3] would not work without the users, who assign tags to the shared bookmarks, images, and music respectively, because tag assignments are used as information source to provide diverse features such as recommendation, search, or exploration features. For example, tag clouds, which depict the popularity of tags within the system, intuitively allow users to explore a repository of tag-annotated resources, just by clicking on tags.

*Corresponding author. Email: abel@l3s.de
[1]http://delicious.com
[2]http://flickr.com
[3]http://last.fm

ISSN: 1361-4568 print/ISSN 1740-7842 online

DOI: 10.1080/1361456YYxxxxxxxx
http://www.informaworld.com

Beside search algorithms that simply detect resources directly annotated with the *search tag*, more advanced algorithms are available that exploit the full structure of the *folksonomy* (Vander Wal 2007). A folksonomy is basically a collection of all tag assignments (user-tag-resource bindings) in the system. It can be modeled as a graph which makes it possible to apply graph-based search and ranking algorithms according to the paradigm of PageRank (Page *et al.* 1998). Such ranking algorithms like FolkRank (Hotho *et al.* 2006c), which is based on PageRank and applicable to folksonomies, not only allow to rank resources but also *tags* and *users*. This feature expands the scope of applications to tag recommendations, user/expert search, etc.

Hence, ranking algorithms play a central role in a multitude of applications, however all ranking algorithms have to face the problem of ambiguity. For example, the tag "java" might be assigned to resources related to programming or the island of Indonesia. Another problem is caused by tags that are re-used on various occasions with different (though implicit) meaning. For instance, the tag "to-read" might be added by a same user at different times to scientific papers that are relevant for a research work or to websites that explain what to see in some location the user would like to visit on holidays. If the tag "to-read" would be used in a query, likely the ranking algorithm outcome would not satisfy the user because such algorithms lack the means to contextualize the ranking. Correspondingly, for broad tags like "music" or "web", which are assigned to a huge amount of resources, it is difficult to compute a ranking that fits to the actual desires of the user.

In this article we examine how the problems, mentioned above, can be solved by exploiting *context information* that is either embedded in the folksonomy or constructed from the user interactions (*clicks*). We present a *lightweight approach for contextualized search in folksonomy systems* without requiring the systems to do extensive user modeling and without any prerequisites for the user. We do so by proposing general strategies that rely on the context information in a way that is *orthogonal* to the ranking algorithm that is used. We model context by the notion of tag clouds (list of weighted tags), e.g., it can be formed by the tag cloud of a resource, from which the user initiates a search activity. So, for example, if a user has navigated to an image in Flickr showing the Indonesian island Java and thereafter clicks on the tag "java" to explore further photos of the island, then it is beneficial to consider also the other tags of the image (e.g. "indonesia", etc.) to adapt the outcome of the search to the user's actual needs.

The interesting novelty of our proposal is that we do not only restrict the context information to the profile of the user, who initiates the query, but also present strategies that consider *content* the user is currently browsing. In addition to the contextual browsing we analyze the benefits of *embedding context information directly into the folksonomy*. Therefore, we present a model that enables systems to attach context information explicitly to particular tag assignments. For our investigations we implemented TagMe! (Abel *et al.* 2009c), a tagging and exploration interface for Flickr pictures, that introduces three types of context: (i) spatial information describing to which part of a resource a tag assignment belongs to, (ii) categories for organizing tag assignments, and (iii) URIs that describe the semantic meaning of a tag assignment. Our experiments reveal that the exploitation of such context information has a significant impact on the search performance.

In this article we introduce different approaches for leveraging context in folksonomy systems. The main contributions can be summarized as follows.

- We introduce a lightweight approach that models context by means of tag clouds and allows the adaption of search results, produced by arbitrary ranking algorithms, to the actual user needs.
- We propose a new folksonomy model for embedding arbitrary context informa-

tion into folksonomies and show that the consideration of context significantly improves existing ranking algorithms.

- We present new ranking algorithms such as SocialHITS, a novel ranking algorithm, which adapts the HITS algorithm (Kleinberg 1999) to folksonomies, as well as FolkRank-based algorithms and prove that there are situations in which our algorithms significantly perform better than existing ranking algorithms.

In our experiments, we not only evaluate the quality of ranking resources and tags, but also—and this constitutes the originality of our experiments—measure the performance of ranking user entities. Further, we showcase our approaches in two different systems: In the TagMe! system (Abel *et al.* 2009c), a tagging and exploration front-end for Flickr, as well as in the GroupMe! system (Abel *et al.* 2007), a social bookmarking system that allows users to organize their bookmarks in thematic groups.

The article is organized as follows. In the next section we will discuss previous work and provide further motivation for the work presented in this article. In Section 2 we present our notion of context in folksonomy systems. We define a formal context folksonomy model and outline our approach to contextualize rankings. In Section 3 we discuss several ranking algorithms that exploit context information in folksonomy systems. We evaluate our approaches with respect to different applications: In Section 4 we evaluate our approaches for adapting search to the actual desires of individual users and in Section 5 we examine how context embedded in the folksonomy can help to improve search. We end our paper with conclusions in Section 6. Section 4 is based on work that we presented in (Abel *et al.* 2009a).

1.1 Related Work

For research carried out in the field of tagging systems the understanding of folksonomies (Vander Wal 2007), which evolve over time when *users* assign *tags* to *resources*, is a matter of particular interest. Formal folksonomy models have been proposed in (Hotho *et al.* 2006a, Mika 2005) and usually interpret a folksonomy as collection of tag assignments possibly enriched with context information like time (Halpin *et al.* 2007) or characteristics of the setting, in which a tag assignment was made (Abel *et al.* 2007). Further, there exist models that try to incorporate the tagging behavior of users (Dellschaft and Staab 2008, Halpin *et al.* 2007). We extend those folksonomy models with context information that can be attached to individual tag assignments. For example, we follow the MOAT (Passant and Laublet 2008) approach and attach DBpedia URIs (Auer *et al.* 2007), which refer to the structured Wikipedia data, to clearly define the meaning of tag assignments.

Given traditional folksonomy models there are a number of research fields including but not limited to the design of search algorithms (Bao *et al.* 2007, Hotho *et al.* 2006c), computing recommendations (Byde *et al.* 2007, Jäschke *et al.* 2007, Sigurbjörnsson and van Zwol 2008), deducing semantics from tags (Hotho *et al.* 2006b, Rattenbury *et al.* 2007), or user modeling (Firan *et al.* 2007, Li *et al.* 2008, Michlmayr and Cayzer 2007). Here, ranking algorithms are indispensable as they allow for the ordering of search results, recommendations, etc. A fundamental assumption of research in the field of folksonomy systems is that tags assigned by the users to resources describe the content of the resources very well. This assumption is proved in (Li *et al.* 2008), where the authors compared the actual content of Web pages with tags assigned to these pages in the Delicious system.

In (Abel *et al.* 2009b) we compared different ranking algorithms, FolkRank (Hotho *et al.* 2006c), GFolkRank (Abel *et al.* 2009b), GRank (Abel *et al.* 2009b), SocialPageRank, and SocialSimRank (Bao *et al.* 2007), with regards to their ability to rank resources and tags. We discovered that those algorithms which utilize the entirety of the information in a folksonomy (including context information attached to the tag assignments) performed best. Our previous findings motivate the work presented in this article. We propose strategies enabling the integration of context information independently from both the used ranking algorithm and the underlying folksonomy model. From a more technical perspective, our strategies for contextualizing search result rankings are based on *query expansion* (Voorhees 1994). Instead of applying co-occurrence based techniques (Kim and Choi 1999) or using dictionaries, such as WordNet, we follow the approach of (Chirita *et al.* 2007) and utilize context information to expand queries and contextualize rankings.

Previous work related to ranking in folksonomies mainly focusses on ranking of resources (Abel *et al.* 2009b, Bao *et al.* 2007, Hotho *et al.* 2006c) or tags (Abel *et al.* 2008, Sigurbjörnsson and van Zwol 2008). In this paper we go beyond state of the art and evaluate folksonomy-based algorithms with respect to their performance when ranking *user entities*, so to allow the identification of users with certain interests or expertise. This capability, though not yet exploited sufficiently in existing social networking services, like Facebook[1] or LinkedIn[2], is of tremendous interest to research in social networking and has many practical applications. While social networking systems require the users to input their interests, competencies, or relations to other users explicitly, tagging systems, on the contrary, capture such information automatically and allow social networks to be constructed implicitly (Nauerz and Groh 2008). However, the retrieval of user entities has not been studied sufficiently in the field of folksonomy systems. Hence, we evaluate only ranking algorithms, which also allow the ranking of users. Based on Kleinberg's Hyperlink-Induced Topic Search (HITS) algorithm (Kleinberg 1999) and ideas presented in (Wu *et al.* 2006) we propose SocialHITS, as the notion of *authorities* and *hubs* appears to be appropriate for user entities, in particular.

2. Context in Folksonomies

Folksonomies evolve over time when users annotate resources with freely chosen keywords. Research in the area of folksonomy systems most often focusses on lightweight models where folksonomies are basically considered as collections of tag assignments. In this section we introduce approaches for modeling context in folksonomy systems. We present strategies for embedding context information in formal folksonomy models (see Section 2.1) and propose approaches for constructing context from user interactions.

2.1 Folksonomy Models

Traditional folksonomy models describe the relations between users, tags and resources. According to (Hotho *et al.* 2006a), a folksonomy can be defined as follows.

Definition 2.1: A *folksonomy* is a quadruple $\mathbb{F} := (U, T, R, Y)$, where U, T, R are finite sets of instances of *users*, *tags*, and *resources*. Y defines a relation, the *tag assignment*, between these sets, that is, $Y \subseteq U \times T \times R$.

Some systems imply a folksonomy model that incorporates additional information indicating in which context a tag was assigned to a resource. In particular,

[1]http://facebook.com
[2]http://linkedin.com

such context might be formed by groups, which are finite sets of resources. The corresponding *group context folksonomy* is defined in (Abel *et al.* 2009b) as follows.

Definition 2.2: A *group context folksonomy* is a 5-tuple $\mathbb{F} := (U, T, \breve{R}, G, \breve{Y})$, where U, T, R, G are finite sets that contain instances of users, tags, resources, and groups, respectively. $\breve{R} = R \cup G$ is the union of the set of resources and the set of groups and $\breve{Y}$ defines a tag assignment having a *group context*: $\breve{Y} \subseteq U \times T \times \breve{R} \times (G \cup \{\varepsilon\})$, where ε is a reserved symbol for the *empty group context*, i.e. if there is no group context available.

Group context folksonomies evolve in systems like GroupMe![1] (Abel *et al.* 2007), which allows for tagging of bookmarks in the context of a group of related bookmarks, or Flickr, which enables users to create sets of images they can tag. In Definition 2.2, the group context is attached to the tag assignments. In Definition 2.3 we introduce a more generic folksonomy model that allows us to attach arbitrary type of context to tag assignments.

Definition 2.3: A *context folksonomy* is a tuple $\mathbb{F} := (U, T, R, Y, C, Z)$, where:

- U, T, R, C are finite sets of instances of *users*, *tags*, *resources*, and *context information* respectively,
- Y defines a relation, the *tag assignment* that is, $Y \subseteq U \times T \times R$ and
- Z defines a relation, the *context assignment* that is $Z \subseteq Y \times C$

Given the context folksonomy model, it is possible to attach any kind of context to tag assignments. For example, the model allows for tagging tag assignments. TagMe![2], a tagging and exploration front-end for Flickr pictures, introduces three types of context: (i) spatial information describing to which part of a resource a tag assignment belongs to, (ii) categories for organizing tag assignments, and (iii) DBpedia URIs that describe the semantic meaning of a tag assignment. Such context information is simply assigned to a tag assignment by the relation Z.

The spatial information as well as the categories are explicitly provided by the end-users via the tagging interface of TagMe! (see Figure 1). For each tag assignment a user can enter one or more categories that classify the annotation. While typing in a category, the users get auto-completion suggestions from the pre-existing categories of the user community (see bottom in Figure 1). When a user categorizes a tag assignment $y = (u, t, r) \in Y$ into category c then this is modeled as relation $(y, c) \in Z$ where $c \in C$ can actually be an arbitrary tag, i.e. $c \in T$. Further, users are enabled to perform *spatial tag assignments*, i.e. to attach a tag assignment to a specific area, which they can draw within the picture (see rectangle within the photo in Figure 1) similarly to *notes* in Flickr or annotations in LabelMe (Russell *et al.* 2008). In the context folksonomy a spatial tag assignment is simply modeled via a relation $(c_a, y) \in Z$ where c_a refers to the context information that describes the area that is tagged. TagMe! automatically assigns DBpedia URIs to tag assignments by exploiting the DBpedia lookup service[3] (cf. Section 5.1.3). Hence, all tag assignments have well-defined semantics so that applications, which operate on TagMe! data, can clearly understand the meaning of the tag assignments.

[1] http://groupme.org
[2] http://tagme.groupme.org
[3] http://lookup.dbpedia.org

Figure 1. User tags an area within an image and categorizes the tag assignment with support of the TagMe! system.

2.2 *Inferring Context from User Interactions*

The group and context folksonomies presented above require users to perform additional steps that go slightly beyond traditional tagging. In this section we examine how context information can be deduced from user interactions that do not necessarily require additional inputs. To give some intuition for the notion of context inferred from user interactions in folksonomy systems, we first describe a characteristic scenario in the GroupMe! tagging system, which we also used as test environment to conduct our experiments in the scope of contextualized browsing (see Section 4). GroupMe! (Abel *et al.* 2007) enables users to manage their bookmarks and share them with other users and allows users to organize bookmarks in groups. Bookmarks as well as the groups can be annotated with tags.

2.2.1 *Scenario*

Let us consider that Bob is planning to travel to the Hypertext conference 2009. Therefore, he creates a GroupMe! group entitled "Trip to Hypertext '09, Turin", in which he adds bookmarks referring to the conference website or to some video showing sights of Turin. He also annotates his bookmarks with tags like "hypertext", "2009", or "conference" to facilitate future retrieval (cf. Figure 2). Bob would appreciate some tag suggestions that expedite the tagging process. Alice is browsing through the GroupMe! system and stumbles upon Bob's group, because she is interested in submitting a paper to that conference. However, via the bookmarked conference website, which is part of the group, she finds out that the deadline has already passed. She now clicks on the tag "conference" and when she does so, likely she is not interested in *any* conference but in conferences that are related either to the same topics or to the year 2009 or that are related to combinations of all such features. Furthermore, she would be delighted to find expert users with whom she could discuss about appropriate conferences and corresponding topics.

In the scenario, the consideration of context can help to improve the usability of the tagging system: when computing tag suggestions, Bob's user profile as well as the tags that have already been assigned to other bookmarks in the "Trip to Hypertext '09, Turin" group can be considered. Further, when Alice clicks on the

Figure 2. GroupMe! group about traveling to Hypertext conference 2009 in Turin.

tag "conference" she neither wants to retrieve bookmarks related to conferences in the field of biology nor seeks for information about past conferences, but she would like to obtain content relevant to computer science conferences in 2009. To adapt the search result to Alice's needs it would be appropriate to include the tags that occur within the Web page Alice visited when she clicked on "conference". Hence, adaptation would even be possible if Alice is not known to the tagging system or if she rarely interacts with the system so that the system has no detailed profile of Alice yet.

2.2.2 Constructing Context

Our approach models contextual user interactions in folksonomy systems as tag clouds, which are lists of weighted tags. Based on the traditional folksonomy model specified in Definition 2.1, a tag cloud can be computed for users, tags, and resources by counting tag assignments, e.g. the tag cloud of a user can be defined as follows.

Definition 2.4: The tag cloud $TC_U(u)$ of a user u is $TC_U(u) = \{\{t, w(u,t)\} | (u,t,r) \in Y, \; w(u,t) = |\{r \in R : (u,t,r) \in Y\}|\}$, where $w(u,t)$ is the number of tag assignments where user u assigned tag t to some resource r.

Hence, the weight assigned to a tag simply corresponds to the usage frequency of the tag. We normalize the weights so that the sum of the weights assigned to the tags in the tag cloud is equal to 1. Furthermore, we use $TC_U@k(u)$, $TC_T@k(t)$, and $TC_R@k(r)$ respectively to refer to the tag cloud that contains only the top k tags, which have the highest weight.

In our scenario, Alice and Bob are acting in the GroupMe! system, which implies a *group context folksonomy* (see Definition 2.2). Given such a group context folksonomy, tag clouds for users, tags, and resources are computed correspondingly to traditional folksonomies, whereas a *group tag cloud* $TC_G(g)$ ($g \in G$) is computed by unifying $TC_R(g)$ (groups are resources as well and can therefore be tagged) and the tag clouds of resources contained in g. In this article, we compare three lightweight approaches for constructing context from user interactions.

- **user** The user context is the top k tag cloud ($TC_U@k(u)$) of the user, who is acting and whose actions should be contextualized, i.e. the tags he/she used most frequently.

- **resource** If a user has navigated to a certain resource r then the tag cloud of the resource $TC_R@k(r)$ can be used as context to adapt to his/her next activities.
- **group** Correspondingly, if the user currently browses a group g of resources, e.g. a GroupMe! group or a set of images in Flickr, then $TC_G@k(g)$ can model his/her context.

The user context corresponds to the *naive user modeling strategy* described in (Michlmayr and Cayzer 2007) and only works if the user is already known to the system by means of previously performed tagging activities. In our evaluation we utilize the user context strategy as the benchmark and investigate whether the resource and group context strategies, which do not require any previous knowledge about the user, can compete with the user context strategy.

The context models are deliberately simple. More complex models can be constructed by combining the context models above or by logging resource and group context for a user over a specific period in time. In our evaluation in Section 4 we set $k = 20$ and thus considered the top 20 tags of the tag clouds.

2.2.3 Contextualizing Rankings

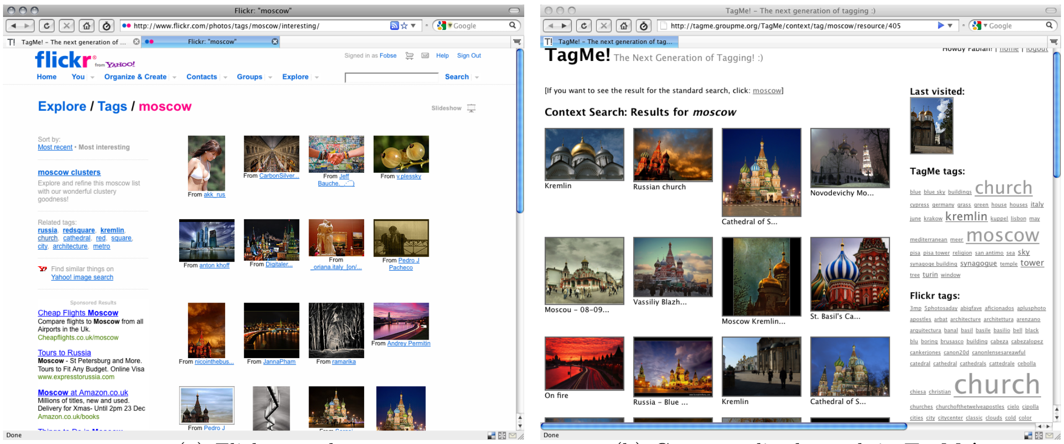
Our approach of inferring context from user interaction targets topic-sensitive ranking algorithms, i.e. algorithms that rank entities (users, tags, and resources) with respect to some topic specified via a query. With *contextualization of rankings* we mean that the ranking respects the query as well as the context given by means of a tag cloud (see previous section). In the scenario above the query was given as single tag, e.g. Alice clicked on a tag to retrieve both, a ranked list of resources and a ranked list of users, who are experts in Alice's current area of interest. A query might however also consist of multiple tags and can therewith be interpreted as tag cloud as well, where the tags are usually weighted equally.

Definition 2.5: The *generic algorithm for computing contextualized rankings* simply combines the ranking computed with respect to the query tag cloud with the one computed for the context tag cloud.

(1) **Input:** query TC_q, context TC_c, folksonomy $\mathbb{F}$, ranking algorithm a, context influence $d \in [0..1]$.
(2) Compute a ranking R_q based on the query tag cloud, $R_q \leftarrow a.rank(TC_q, \mathbb{F})$, and a ranking R_c based on the context tag cloud, $R_c \leftarrow a.rank(TC_c, \mathbb{F})$. R_q and R_c are sets of weighted entities (e_i, w_q) and (e_i, w_c) respectively.
(3) Compute the result ranking R_r by averaging R_q and R_c. R_r contains weighted entities $(e_i, w_{i,r})$, where $w_{i,r} = (1-d) \cdot w_{i,q} + d \cdot w_{i,c}$ and d specifies the influence of the ranking scores computed via the context tag cloud.
(4) **Output:** R_r, the set of weighted entities $(e_i,\ w_{i,r})$, where $w_{i,r}$ denotes the weight (ranking score) assigned to the ith entity (user, tag, or resource).

A contextualized ranking is thus the weighted average of the query and context ranking. In TagMe! we apply our approach to contextualize search for pictures so that end-users can immediately experience contextualized browsing. Figure 3 shows a comparison between Flickr search and the contextualized search in TagMe!. In both settings the user is searching with the tag "moscow" as the given query and in both settings the Flickr *interestingness approach*[1], which considers clicks, comments as well as tags in order to determine the interestingness of a picture

[1]http://www.flickr.com/explore/interesting/



(a) Flickr search (b) Contextualized search in TagMe!

Figure 3. Searching for pictures related to "moscow" – Flickr ranking according to *interestingness* vs. contextualized ranking in TagMe!

with respect to a query, is utilized as ranking algorithm. However, TagMe! applies the algorithm for computing contextualized rankings, i.e. it queries Flickr first for pictures related to "moscow", then utilizes the context (TC_c)—and particularly the tag cloud of the last visited resource—to retrieve related Flickr pictures and finally combines both rankings. In the example depicted in Figure 3(b), the user accessed an image showing a church in Moscow Kremlin before clicking on the tag "moscow". TagMe! successfully adapts the resulting search ranking of Flickr pictures to that context as it ranks those pictures higher that are related to both, the search tag ("moscow") and the context tags (e.g., "church", "kremlin").

While the contextualized search and exploration interface of TagMe! is rather a showcase of our contextualization approach, we evaluate the approach extensively in Section 4. In the next section we therefore present different ranking algorithms applicable to the contextualization algorithm specified above.

3. Ranking Algorithms

In this section we present the ranking algorithms that we apply in our experiments (Section 4 and 5) to reveal the benefits of exploiting context information. We first outline the FolkRank (Hotho *et al.* 2006c) algorithm, which we use as baseline strategy for ranking resources in our search experiments. In Section 3.2 we present different algorithms (mostly based on FolkRank) that leverage context folksonomies to improve ranking quality. Finally, we present SocialHITS, an alternative algorithm that can be applied to both traditional folksonomies as well as context folksonomies and promises to be more appropriate for ranking user entities.

3.1 FolkRank

The FolkRank algorithm (Hotho *et al.* 2006c) operates on the folksonomy model specified in Definition 2.1. FolkRank transforms the hypergraph that is spanned by the tag assignments into a weighted tripartite graph $\mathbb{G}_{\mathbb{F}} = (V_{\mathbb{F}}, E_{\mathbb{F}})$, where an edge connects two entities (user, tag, or resource, i.e. $V_{\mathbb{F}} = U \cup T \cup R$) if both entities occur together at a tag assignment within the folksonomy: $E_{\mathbb{F}} = \{\{u,t\},\{t,r\},\{u,r\} | (u,\ t,r) \in Y\}\}$. The weight of an edge corresponds to the amount of the entities' co-occurrences. For example, the weight of an edge connecting a tag t and a resource r is defined as $w(t,r) = |\{u \in U : (u,t,r) \in Y\}|$ (cf.

Definition 2.1) and thus corresponds to the number of users, who have annotated r with t. The constructed graph $\mathbb{G}_\mathbb{F}$ serves as input for an adaption of the Personalized PageRank (Page *et al.* 1998): $\vec{w} \leftarrow dA_{\mathbb{G}_\mathbb{F}}\vec{w} + (1-d)\vec{p}$, where the adjacency matrix $A_{\mathbb{G}_\mathbb{F}}$ models the folksonomy graph $\mathbb{G}_\mathbb{F}$, $\vec{p}$ allows to specify preferences (e.g. for a tag) and d enables to adjust the influence of the preference vector. FolkRank applies the adapted PageRank twice, first with $d = 1$ and second with $d < 1$ (in our evaluation we set $d = 0.7$ as done in (Hotho *et al.* 2006c)). The final vector, $\vec{w} = \vec{w}_{d<1} - \vec{w}_{d=1}$, contains the *FolkRank* of each folksonomy entity.

3.2 Ranking Algorithms for Context Folksonomies

3.2.1 GFolkRank

GFolkRank (Abel *et al.* 2009b) is a context-sensitive ranking algorithm that is based on FolkRank. It expects a *group context folksonomy* (see Def. 2.2) as input and adapts the process of transforming the hypergraph—spanned by the folksonomy—into the weighted folksonomy graph $\mathbb{G}_\mathbb{F}$ (cf. Section 3.1). It interprets groups as artificial tags and creates new tags $t_g \in T_G$, $T_G \cap T = \emptyset$, for each group g. These artificial tags are assigned to all resources contained in g, whereby the user who added a resource to the group, is declared as the *tagger*. The set of nodes is thus extended by T_G: $V_{\mathbb{F}_{new}} = V_\mathbb{F} \cup T_G$. The edges added to $V_\mathbb{F}$ by the GFolkRank algorithm are: $E_{\mathbb{F}_{new}} = E_\mathbb{F} \cup \{\{u, t_g\}, \{t_g, r\}, \{u, r\} | u \in U, t_g \in T_G, r \in \breve{R}, u$ has added r to group $g\}$. We use a constant value w_c to weight those edges because a resource is usually added only once to a certain group.

3.2.2 GRank

GRank (Abel *et al.* 2009b) is a group-sensitive ranking algorithm as well as GFolkRank. However, GRank is not based on FolkRank, but exploits *group context folksonomy* in a straightforward way. Given a query tag t_q, the GRank algorithm detects a set of tag assignments $(u, t, r, g) \in \breve{Y}_q$, where the resource $r \in \breve{R}$ is (a) directly annotated with t_q, (b) contained in a group that is tagged with t_q, (c) grouped together with a resource directly annotated with t_q, or (d) a group which contains a resource directly annotated with t_q. The entities (users, tags, and resources) are then weighted according to their occurrence frequency within the tag assignments of $\breve{Y}_q$. For more details on GRank we refer the reader to (Abel *et al.* 2009b).

3.2.3 Category-based FolkRank

The category-based FolkRank algorithm operates on a context folksonomy (see Definition 2.3) where the context is given by categories that are attached to tag assignments. The algorithm relates folksonomy entities via the category assignments and the main hypothesis is that entities sharing the same category are related to each other. Similarly to GFolkRank, the category-based FolkRank introduces an alternative approach for creating the weighted folksonomy graph $\mathbb{G}_\mathbb{F}$ (cf. Section 3.1). Categories are treated as tags ($c \in T_C$ where $T_C \subseteq T$) so that the set of nodes is extended with T_C: $V_{\mathbb{F}_{new}} = V_\mathbb{F} \cup T_C$. For each category assignment $(y, c) \in Z$, new edges are created to connect the given category c with the resource and tag of the tag assignment y: $E_{\mathbb{F}_{new}} = E_\mathbb{F} \cup \{\{c, r\}, \{c, t\} | c \in T_C, t \in T, r \in R, ((u, t, r), c) \in Z\}$. The weight of an edge (c, r) corresponds to the frequency the category c is assigned to a tag assignment that refers to r: $w(c, r) = |\{(u, t, r) \in Y : (u, t, r) \in Y, ((u, t, r), c) \in Z\}|$. Weights of (c, t)-edges are accordingly computed by counting the tag assignments that refer to t and are categorized using c.

3.2.4 Area-based FolkRank

While the categories are used to enrich the folksonomy graph with further edges and possibly also with further vertices, the area-based FolkRank merely modifies the weights of edges in $\mathbb{G}_{\mathbb{F}}$ (cf. Section 3.1). In particular, it emphasizes the weight of an edge between a tag t and a resource r (i.e. (t, r)-edges) whenever t and r occur within a tag assignment $(u, t, r) \in Y$ to which spatial context information is attached to. The amplification is based on the size of the corresponding area as well as on the distance of the midpoint of the area to the center of the resource (in our experiments we examine pictures).

- **size** Our hypothesis is that the larger the size of an area the more important is also the corresponding tag for the given resource, i.e. the larger the area that is attached to $(u, t, r) \in Y$ is the more relevant t is for r. The size of an area is measured relatively to the size of the resource. For example, if an area is associated to a tag assignment (u, t, r) and the relative size of the area is $s = 0.4$, i.e. the area covers 40% of the resource, then we use s^{-1} to emphasize the weight $w(t, r)$. As different users might attach differently sized areas to (u, t, r), we use the average size $\bar{s}$ of those areas to finally compute the new weight of (t, r)-edges: $w_s(t, r) = \bar{s}^{-1} \cdot w(t, r)$.
- **distance** The second hypothesis is that tag assignments which are according to the spatial information relevant to the center of a resource are more important for the resource than tag assignments which are associated to the margin of a resource. The distance d from the center of the area to the center of the resource is also measured relatively and the weight $w(t, r)$ is emphasized with the average distance $\bar{d}$ of the areas attached to $(u, t, r) \in Y$: $w_d(t, r) = \bar{d}^{-1} \cdot w(t, r)$.

Finally, the weight of the edges (t, r) is simply the average of $w_s(t, r)$ and $w_d(t, r)$: $w_{area}(t, r) = 0.5 \cdot w_s(t, r) + 0.5 \cdot w_d(t, r)$.

3.2.5 URI-based FolkRank

The URI-based FolkRank operates on meaningful URIs instead of tags. Hence, the construction of the folksonomy graph $\mathbb{G}_{\mathbb{F}} = (V_{\mathbb{F}}, E_{\mathbb{F}})$ is modified as follows. The set of vertices is $V_{\mathbb{F}} = U \cup URI \cup R$, where $URI \subseteq C$ (cf. Definition 2.3) denotes the set of URIs that describe the meaning of the tag assignments. The set of edges is $E_{\mathbb{F}} = \{\{u, uri\}, \{uri, r\}, \{u, r\} | u \in U, uri \in URI, r \in R, ((u, t, r), uri) \in Z\}$ whereas there should only exists exactly one URI assignment $(y, uri) \in Z$ for each tag assignment y. The weights of the edges are computed in the same way as done by the traditional FolkRank algorithm.

The URI-based FolkRank algorithm is resistant against ambiguous tags as well as synonymic tags. For example, given two tag assignments $y_1 = (u_1, t_1, r_1)$ and $y_2 = (u_2, t_2, r_2)$ as well as two context assignments (y_1, uri_1) and (y_2, uri_1), the URI-based FolkRank algorithm would replace the synonymic tags t_1 and t_2 by the unique URI uri_1 that clearly defines the meaning of the tags. It therewith, e.g., relates r_1 and r_2 as it constructs the edges (uri_1, r_1) and (uri_1, r_2). As the TagMe! system utilizes DBpedia URIs to define the meaning of tags, we denote the URI-based FolkRank as *DBpedia FolkRank* in our search experiments in Section 5.

3.3 SocialHITS

In (Kleinberg 1999) the author introduces the *HITS* algorithm that enables to detect hub and authority entities in hyperlinked network structures. A *hub* describes an entity that links to many high quality authority entities and an *authority* denotes an entity, which is linked by many high quality hub entities. Hence, the

HITS algorithm is based on a mutually reinforcing relationship between hubs and authorities. Therefore, the operations that update the authority weight $x^{\langle p \rangle}$ and hub weight $y^{\langle p \rangle}$ of an entity p are defined by the operations A and H (Kleinberg 1999).

$$A : x^{\langle p \rangle} \leftarrow \sum_{q:(q,p) \in E} y^{\langle q \rangle} \tag{1}$$

$$H : y^{\langle p \rangle} \leftarrow \sum_{q:(p,q) \in E} x^{\langle q \rangle} \tag{2}$$

Here, E denotes the set of directed edges within the given graph G. The core algorithm of HITS, which detects the authorities and hubs in a given graph G, performs k iterations in order to update $x^{\langle p \rangle}$ and $y^{\langle p \rangle}$ for each entity (node) within G. The core iteration is defined as follows (Kleinberg 1999).

Definition 3.1: Core HITS iteration.

function iterate(G, k)
G: a graph containing n linked entities
Let x and y be vectors containing the authority and hub weights.
Set x_0 and y_0 to $(\frac{1}{n}, \frac{1}{n}, \frac{1}{n}, ...) \in \mathbb{R}^n$
for $i = 1, 2, ..., k$ **do**:
    $x_i' \leftarrow$ apply A to (x_{i-1}, y_{i-1})
    $y_i' \leftarrow$ apply H to (x_i', y_{i-1})
    $x_i \leftarrow ||x_i'||_1$
    $y_i \leftarrow ||y_i'||_1$
end
return (x_k, y_k)

The graph G that is passed to the core iteration of HITS has to be a directed graph. In general, G is a partial Web graph consisting of linked resources that are possibly relevant to a certain topic (cf. (Kleinberg 1999)). The challenge of applying HITS to folksonomies is to transform a folksonomy into a directed graph in contrast to an undirected graph ($\mathbb{G}_{\mathbb{F}}$) as done by the ranking algorithms in the previous sections. The tag assignments do not explicitly prescribe a direction. In (Wu *et al.* 2006) the authors propose the following strategy: If there is a tag assignment $(u, t, r) \in Y$ then the edges *"$u \rightarrow t$"* and *"$t \rightarrow r$"* will be constructed. Hence, hubs are restricted to be users while the authority role is bound to resources. In our evaluations we will denote that strategy as *naive HITS*. Our approach does not limit the role of hubs and authorities to certain folksonomy entity types, but makes it possible to detect authorative users as well.

The construction of the *directed folksonomy graph* has to consider the design of the folksonomy system and its user interface in particular. In the GroupMe! system, for example, a resource r_h can be interpreted as a hub of a tag t_a assigned to r_h because each resource displays its tag cloud, whereas in tagging systems that do not show the tags of resources it is not possible to draw that conclusion (cf. *tagging support: "viewable" vs. "blind"* in (Marlow *et al.* 2006a)).

Table 1 lists some of the characteristics of users, tags, and resources that indicate when they should be considered as authorities and hubs respectively. Some of these characteristics can be deduced from the traditional folksonomy model (see Definition 2.1) while others require additional context information, e.g. regarding user entities, edges representing some user characteristics can be constructed as follows.

Table 1. Overview of some characteristics of authority/hub users, tags, and resources.

Authorities	
user	a high quality user annotates high quality resources before other users annotate them
tag	is assigned by high quality users
resource	(1) is tagged by high quality users with high quality tags (2) is contained in high quality groups
Hubs	
user	has annotated high quality resources and utilized high quality tags
tag	is assigned to high quality resources
resource	(1) is tagged with tags of high quality resources (2) is contained in groups with high quality resources

- **hub users** For all resources r a user u has annotated with a tag t we can construct edges "$u \rightarrow t$" and "$u \rightarrow r$". The required information is thus contained in the tag assignments.
- **authority users** According to Table 1 an authorative user u_a can also be characterized by the fact that other users have annotated resources, that u_a has annotated before the other users annotated them. Therefore, the timestamp of tag assignments has to be evaluated so that we can construct an edge "$u_h \rightarrow u_a$" whenever another user u_h has annotated a resource that was already tagged by u_a.

Having an appropriate strategy for constructing the directed folksonomy graph, which serves as input to the core HITS iteration (see Definition 3.1), SocialHITS can be defined as follows.

Definition 3.2: The *SocialHITS* algorithm computes hub and authority values for arbitrary folksonomy entities.

(1) **Input:** folksonomy $\mathbb{F}$, topic t, search strategy s_t, graph construction strategy s_g, and the number of HITS iterations k to perform
(2) $\mathbb{F}_t \leftarrow$ apply s_t to $\mathbb{F}$ in order to search for entities and tag assignments relevant to t
(3) $\mathbb{G}_D \leftarrow$ apply s_g to $\mathbb{F}_t$
(4) $(x_k, y_k) \leftarrow \text{iterate}(\mathbb{G}_\mathbb{D}, k)$
(5) **Output:** the vectors x_k and y_k containing the authority and hub values of the entities in $\mathbb{F}_\mathbb{t}$

In our search evaluations we applied a search strategy, which simply accumulated the set of entities delivered by FolkRank, GFolkRank, and GRank (without ranking the items), and utilized the sum of authority and hub score to rank.

4. Contextualized Browsing in Folksonomy Systems

In Section 2.2.2 we proposed different ways to construct context from user interactions by means of tag clouds that describe the actual setting of the user. Section 2.2.3 explained how rankings can be adapted to such context independent of the underlying ranking algorithm. Several applicable ranking algorithms were discussed in Section 3. In summary, we now have a *tool box* that helps tagging systems to adapt rankings to the actual desires of the users. In this section we evaluate the tool box with respect to the following task.

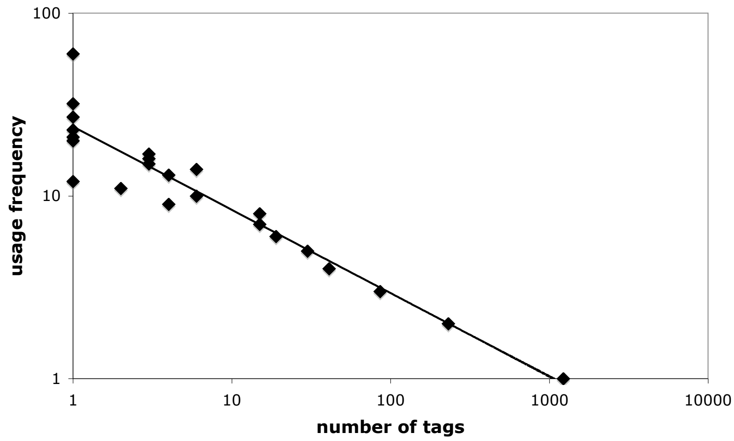


Figure 4. Tag usage in the GroupMe! data set on a logarithmic scale. Only a few distinct tags have been used frequently while most of the tags are only used once.

Ranking Task. *Given a keyword query (tag) and a context (set of weighted tags), the task of the ranking strategy is to compute a ranking of folksonomy entities so that entities that are most relevant to both, the keyword query and the context, appear at the top of the ranking.*

In particular, we will answer the following questions.

(1) How does the consideration of the different context types influence the performance of the algorithms in fulfilling the task above?
(2) Which type of context (cf. Section 2.2.2) is the most appropriate?
(3) Which algorithm (cf. Section 3) performs best with respect to information retrieval metrics such as precision (see Section 4.3)?

We are also interested in the strength of the algorithms regarding the type of entity (user, tag, or resource) that should be ranked. Moreover, the ranking algorithms possibly prefer different types of context. Our goal is to clarify how each individual ranking algorithm can benefit from the knowledge about the context.

4.1 Data Set and Test Set

We run our experiments on a data set of the GroupMe! tagging system (cf. Section 2.2). In the data set we had 450 users, who mainly come from the research community in Computer Science. Together they bookmarked 2189 Web resources, created 550 groups to organize these bookmarks and made 3190 tag assignments using 1699 different tags. Figure 4 illustrates that the tag usage reminds of a power law distribution as there are a lot of tags (72.04%), which were only used once, and only a few tags, which were applied frequently. For example, the tag "semantic web" was assigned 60 times and was therewith the most frequently used tag. Hence, regarding the tag usage distribution we observed similar characteristics as they occur also in larger data sets (cf. (Dellschaft and Staab 2008, Halpin *et al.* 2007)).

For our experiments, we defined a test set of 19 search settings, where each setting was formed by a keyword query (tag) and a context consisting of (i) the *user* u, who performs a search activity, (ii) the *resource* r the user u accessed before initiating the search activity, and (iii) the *group* that contains r. We thus simulated the scenario described in Section 2.2, where the user Alice first accessed a group of resources, which were related to the "Hypertext '09 conference", then

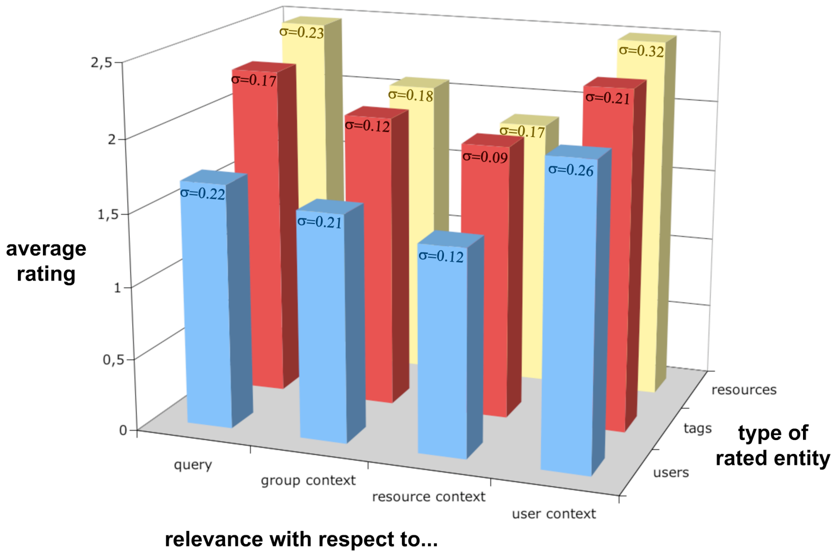


Figure 5. Characteristics of the judgment behavior in the user study with respect to the types of rated entities (user, tag, or resource) and the type of judgment basis (query, group context, resource context, or user context)

focused a certain resource (the conference website), before she finally clicked on the tag "conference" to search for related content. For the search settings, we selected tags as queries that cover the different spectra of the tag usage distribution. In particular, we chose 6 tags that were used 1-10 times (e.g. "soa" and "james bond"), 9 tags that were used 11-20 times (e.g. "conference" and "beer"), and 4 tags that were used more than 20 times (e.g. "hannover" and "semantic web"). The topics of the different search settings represented the diversity of topics available in the GroupMe! data set. For each of the 19 search settings we also selected a resource and a corresponding group as context, where the *resource context tag cloud* (cf. $TC_R(r)$, Section 2.2.2) contained 3.21 tags on average and the *group context tag cloud* $TC_G(g)$ contained 13.58 tags. Further, for each search setting we defined a user as *actor*. Here, the condition was that the actor is also related to the topic of the setting, i.e. we only selected those users who already used the tags that occurred in the tag clouds of the corresponding resource ($TC_R(r)$) and group ($TC_G(g)$) of the setting. Thereby, we tried to give the user modeling strategy ($TC_U(r)$) the same opportunities as the resource and group context strategies to fulfill the task defined above.

4.2 User Study

Given the different search settings, we conducted a user study with users of the GroupMe! system (10 PhD students and student assistants) where the participants had to do relevance assessment (for the given setting, the low number of 10 participants was sufficient to obtain significant results). We presented the participants of the study a search setting together with a list of users, tags, and resources that were determined by accumulating the rankings of the different strategies for the given search setting. For each entity (user, tag, or resource) the participants judged the relevance of the entity with respect to the (i) query, (ii) group context, (iii) resource context, and (iv) user (actor) context. Therefore, they were enabled to easily gather information on which they could constitute their judgements, e.g. all involved entities were clickable and the participants were able to see an entity while judging it. In particular, the participants had to answer whether an entity is relevant or not on a five-point scale: *yes*, *rather yes*, *rather no*, *no*, and *don't know*.

Thereby, we obtained a set of 8593 user-generated judgements, in particular 1550 *yes*, 1549 *rather yes*, 1097 *rather no*, 4242 *no*, and 155 *don't know* judgements. Figure 5 overviews the 8593 user judgements and the overall judging behavior of the participants with respect to the type of entity (user, tag, and resource) that was judged on the basis of its relevance to the query and the different parts of the context (group, resource, and user context). The average judgement is given as number, where 0 means *don't know*, 1 means *no*, 2 means *rather no*, etc. The standard deviation σ is averaged across the deviations of judgments, where the different participants evaluated the same entity with respect to the same query/context.

Overall, the standard deviation indicates that the judgments of the participants were very homogeneous. Rating the relevance of entities with respect to the user context was probably the most difficult task for the participants, because they had to browse the profile of the corresponding user, i.e. the groups he/she created, the resources he/she bookmarked, and the tags he/she used in the past. Hence, the standard deviation for that judgement task is higher than for the others. Judging tags was the most intuitive task and also gained the most homogenous judgements. On average, the resources were rated better than tags, and users. This can be explained by the number of possibly relevant entities listed in the user study. For example, there were probably less than 5 of 22 users but more than 20 of 43 resources relevant to the query "james bond". However, even if there would be a slightly different judging behavior regarding the different types of entities (users, tags, and resources) then this would not influence our results as all algorithms were initialized with the same settings.

In general, the characteristics of the data set of judgements carried out during the user study enable us to gain statistically well-grounded results.

4.3 Method and Metrics

According to the *ranking task*, which we defined at the beginning of the section, the different strategies had to rank users, tags, and resources with respect to a given search setting consisting of a query and context as described in Section 4.1. We combined the ranking algorithms presented in Section 3 that are applicable to group context folksonomies—FolkRank, GFolkRank, GRank, and SocialHITS—with the different context models presented in Section 2.2.2 and then passed them to the algorithm for contextualizing rankings (Definition 2.5 in Section 2.2.3). Thereby we obtained 12 strategies, e.g. *FolkRank(user)*, which denotes the strategy that applies the FolkRank algorithm together with the user context, or *GRank(resource)*, which is the strategy that contextualizes the ranking produced by GRank with the resource context. Each ranking strategy then had to compute a user, tag, and resource ranking for each of the 19 search settings, which consist of a query and the (user, group, and resource) context. Thus, each strategy had to compute 57 rankings.

To measure the quality of the rankings we used the following metrics (cf. (Sigurbjörnsson and van Zwol 2008)):

- **MRR** The *MRR* (Mean Reciprocal Rank) indicates at which rank the first *relevant* entity occurs on average.
- **S@k** The Success at rank k (*S@k*) stands for the mean probability that a *relevant* entity occurs within the top k of the ranking.
- **P@k** Precision at rank k (*P@K*) represents the average proportion of *relevant* entities within the top k.

For our experiment we considered an entity as *relevant* iff the average user judge-

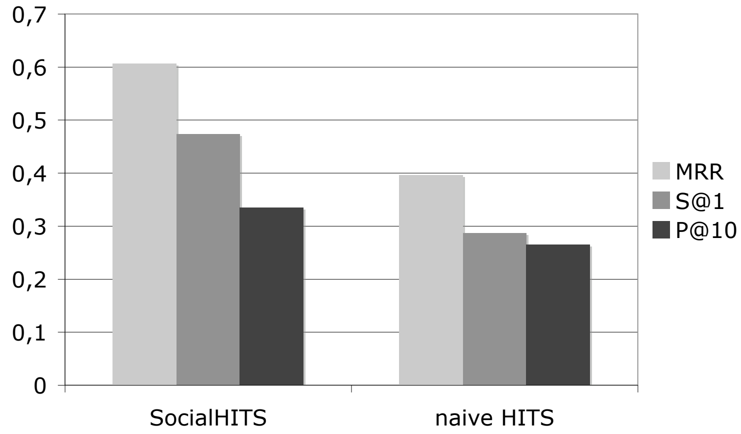


Figure 6. SocialHITS vs. naive HITS strategy (ordered by MRR(both)).

ment is at least "rather yes" (rating score ≥ 3.0), e.g. given three "rather yes" (rating score = 3) judgments and two "rather no" judgments (rating score = 2) for the same entity with respect to some setting then this was treated as *not relevant*, because the average rating score is 2.6 and therewith smaller than 3.0 ("rather yes"). Judgements where the participant stated "don't know" were treated as "no".

4.4 ***Results***

We present the results according to the following structure. We first try to evaluate the performance of the newly introduced SocialHITS algorithm, independently from the used context strategy. Afterwards we overview our core results that allow us to answer the questions raised at the beginning of this section. In Subsection 4.4.3 we analyze the performance of the strategies when they have to rank (a) user and (b) resource entities. We will particularly investigate the ability of the algorithms to rank users, because this has not been studied extensively in previous work yet. Our result analysis finishes with a summary regarding the performance of the different context models, which are used to adapt the rankings to the actual context of a user.

We tested the statistical significance of all following results with a two-tailed t-Test and a significance level of $\alpha = 0.05$. The null hypothesis H0 is that some strategy s_1 is as good as another strategy s_2, while H1 states that s_1 is better than s_2.

4.4.1 SocialHITS vs. naive HITS

The SocialHITS algorithm, which we introduced in Definition 3.2, expects a graph construction strategy as input, which creates a directed graph from the given folksonomy. A naive approach to construct such a graph is presented in (Wu *et al.* 2006). Figure 6 compares this straightforward application of HITS with SocialHITS, a more complex approach, which causes a graph with higher *compactness*. The results are based on 171 test runs, where the algorithms had to rank user, tags, or resources regarding the different search settings described above. Entities were considered as relevant iff they were, according to the user judgments, relevant to both, the query and the context. SocialHITS outperforms the naive HITS algorithm significantly with respect to all metrics. For example, the mean reciprocal rank (MRR), which indicates the average rank of the first relevant entity, is more than 50% better when using SocialHITS instead of the naive approach. The same holds for S@1. In particular, the probability that a relevant entity appears at the first rank is 47.4% when using SocialHITS in contrast to 28.7% when the naive

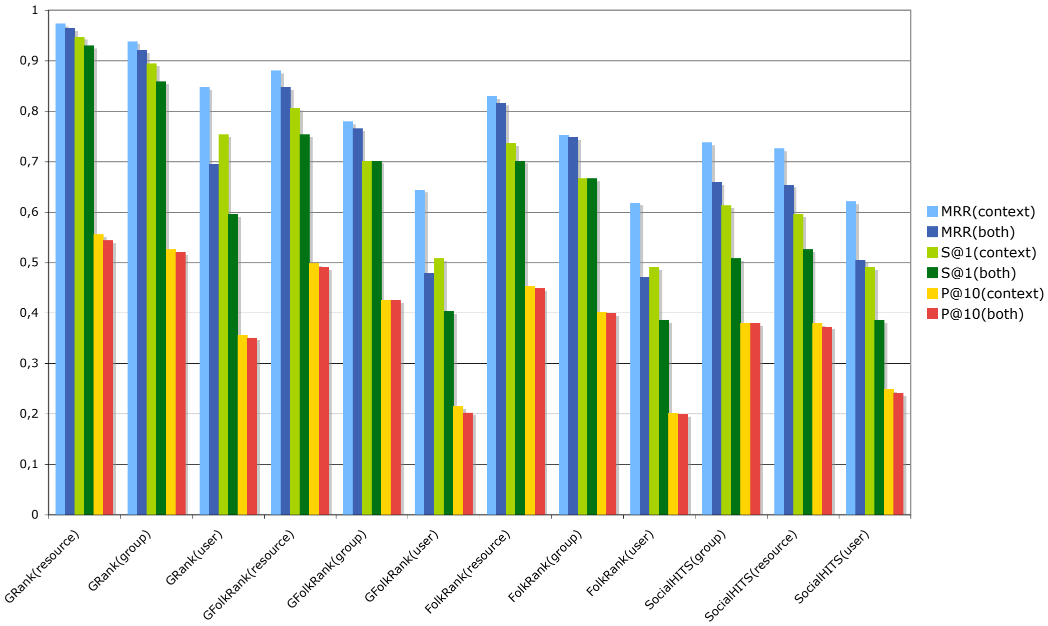


Figure 7. Performance of the different strategies with respect to the task of ranking folksonomy entities (ordered by MRR(both)).

approach is applied. Further, the precision within the top 10 is significantly higher for the SocialHITS algorithm.

The performance differences were obvious for every single ranking result. The naive HITS algorithm performed worst when it had to rank user entities. This can be explained from the underlying graph construction strategy, which implies an authority score of zero for user entities.

As SocialHITS outperforms the naive HITS approach we just consider SocialHITS for our comparisons with the other ranking algorithms presented in Section 3.

4.4.2 Result Overview

Figure 7 overviews the core results of our experiment. It shows the quality of the ranking algorithms (Section 3) in combination with the different context models (Section 2.2.2) when using the contextualization strategy defined in Section 2.2.3. The metrics MRR(context), S@1(context), and P@10(context) determine the relevance of a particular entity with respect to the context, which is formed by the actor of a search setting as well as the resource and group context. For MRR(both), S@1(both), and P@10(both) relevance is given iff the entity is relevant to *both*, the query and the context of a search setting.

The GRank algorithm in combination with the resource context (*GRank(resource)*) is the most successful strategy for computing folksonomy entity rankings that should be adapted to a given search setting. *GRank(resource)* significantly performs better than all other strategies except for *GRank (group)* and *GFolkRank(resource)*. Overall, Figure 7 reveals two main results: (1) the GRank algorithm is the best performing algorithm and (2) independently from the used algorithm, the resource and group context models produce better results than the user context strategy.

It is interesting to see that the precisions P@10(context) and P@10(both) do not differ significantly, which means that the items, which are included into the top 10 rankings because of their relevance to the context, are also relevant to the query. This gives supplemental motivation for the work, presented in this paper, as it indicates that the consideration of context does not reduce the precision of the result rankings within the top 10. Similarly, this motivation can be deduced

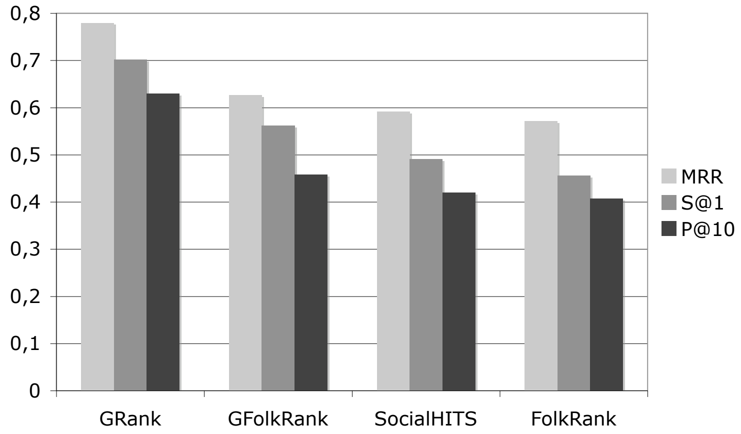


Figure 8. Performance of the different algorithms with respect to the task of ranking resources (ordered by MRR).

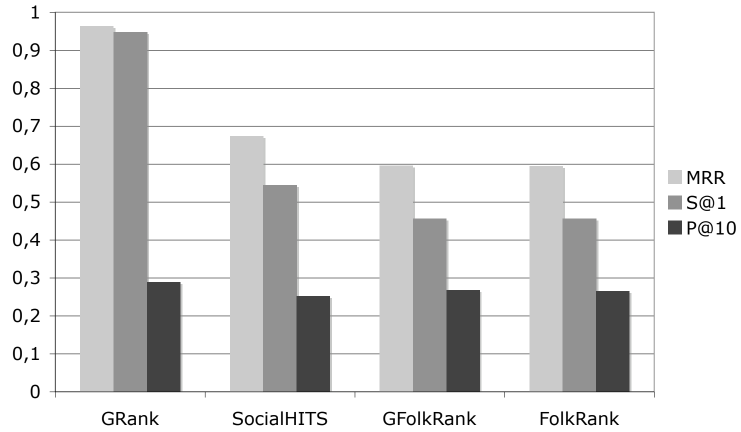


Figure 9. Performance of the different algorithms with respect to the task of ranking user (ordered by MRR).

from the S@1 metrics, as there is no significant difference between S@1(context) and S@1(both) for the strategies that make use of the resource or group context. However, the consideration of user context causes impreciseness regarding *query relevance* at the very top of the ranking. For example, the probability to retrieve an item that is relevant to the context of a search setting is 75.4% when *GRank(user)* is applied, whereas the probability that this item is relevant to the query as well is just 59.6%.

Between FolkRank and GFolkRank, the group-sensitive extension of FolkRank, there is not a significant difference in general, but GFolkRank performs better for all the different context models than FolkRank. The SocialHITS algorithm tends to be outperformed by the other algorithms. The performance of SocialHITS depends on the type of entity that should be ranked, while the performance of the other algorithms is rather constant, in this regard. SocialHITS significantly performs worse when it has to rank tags instead of users or resources. Hence, the role of tags in the model of SocialHITS (cf. Table 1) should possibly be revised in future work to make SocialHITS also applicable to the ranking of tags.

4.4.3 Ranking Users and Resources

The task of ranking resources is possibly the most prominent ranking application, because it is, for example, applied to put search results into an appropriate order. Figure 8 overviews the performance of the different algorithms for that task averaged across the test runs targeting the different search settings while considering

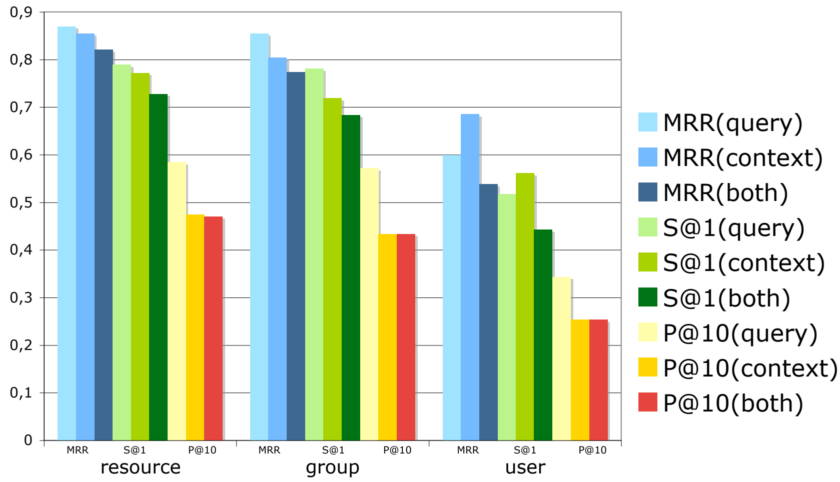


Figure 10. Performance depending on the used context type (ordered by MRR(both).

either the user, group, or resource context. The metrics MRR, S@1, and P@10 are measured based on the relevance of a resource to both, the query and the context of the corresponding search setting.

GRank is significantly the best algorithm to rank resources followed by GFolkRank. Both algorithms exploit group structures in *group context folksonomies* (see Definition 2.2). Such folksonomies arise in tagging systems such as Flickr or GroupMe! which allow their users to group and tag the resources. In folksonomy systems that do not offer the notion of groups these algorithms would not work properly. In these systems SocialHITS would be the preferred choice because it shows better results than the FolkRank algorithm.

The results of the experiment focussing on ranking users is of particular interest because so far there exist – to the best of our knowledge – no studies which analyze the quality of folksonomy-based ranking algorithms in this regard. A set of exiting application can be realized with the aid of an user ranking functionality. For example, it can be applied to find experts on a certain topic or to recommend users to each other, who have – based on their tagging behavior – similar interests.

The qualification of the algorithms to rank user entities can be derived from the results shown in Figure 9. Overall, the outcomes are, regarding P@10, worse than the outcomes of the resource ranking experiment depicted in Figure 8. This can be explained by the absolute number of users possibly relevant to a search setting which is lower in comparison to the number of possibly relevant resources. GRank is again the best performing algorithm. For example, the probability that a user, who is relevant to the query and context, appears in the first position of the ranking is 94.7%. SocialHITS is the second best strategy having S@1 score that is 20% higher than the one of GFolkRank and FolkRank. Further, the mean reciprocal rank (MRR) of SocialHITS is more than 10% better than the one of GFolkRank and FolkRank, which do not differ significantly in their performance. Hence, SocialHITS is again the best choice for settings where no group context exists so that GRank is not applicable.

4.4.4 Synopsis

From the results presented in the previous subsections we can identify GRank, which we introduced in (Abel *et al.* 2009b), as the best performing algorithm for ranking entities in *group context folksonomies*. When it comes to the ranking of users or resources then SocialHITS, which significantly performs better than the naive HITS approach, is the best algorithm operating on the traditional folksonomy model (cf. Definition 2.1).

Figure 10 abstracts from the underlying ranking algorithms and summarizes the results listed in Figure 7 from the perspective of the context type that was considered by the algorithms to adapt the rankings to a particular search setting. According to the results shown in Figure 10, we can clearly put the strategies into an order: (1) the resource context gains significantly better results than the group and user context, (2) the group context strategy produces significantly better results than the user context strategy, while (3) the user context strategy performs worst. As described in Section 2.2.2, context is formed by the tag cloud of a resource, group, or user respectively. The size of the different tag cloud types differed: Resource tag clouds contained on average 3.21 tags, group tag clouds 13.58, and user tag clouds were limited to 20 tags. However, the pure size of the context tag clouds do not only explain the outcomes of the experiment. For example, for some settings group context tag clouds containing more than 15 tags delivered better results than smaller tag clouds while for other settings it was the other way round. Hence, rather the homogeneity of a tag cloud used as context seems to influence the quality of a contextualizing a ranking. The user context, i.e. the top tags of the user who performs a search activity, is thematically multi-faceted, which explains that the mean reciprocal rank measured with respect to the context (MRR(context)) is higher than the MRR measured regarding the relevance to the query (MRR(query)).

Overall, the excellent results of the resource and group context strategies are impressive, because they do not require any previous knowledge about the user, but just capture the current context of a user. The user modeling strategy on the contrary requires such knowledge. Our results have therewith a direct impact on the end users of a tagging system as they can benefit from the adaptation of result rankings to their current needs even if they are not known to the system.

5. Exploiting Semantics of Context Folksonomies

The results presented in the previous section suggest the consideration of context information to improve the browsing experience in folksonomy systems. For example, when users navigate through the resources of a folksonomy it is beneficial to contextualize the tag-based query with the tags of the recently visited resource so that the result list of resources are relevant to both, the query as well as the actual context of the user. The contextualized browsing approach can operate on traditional folksonomies as the context information is constructed from user interactions that are common in traditional tagging systems. In this Section we analyze the benefits of context folksonomies (see Definition 2.3), i.e. context information that is embedded in the folksonomy model. In particular, we analyze the three context types introduced by the TagMe! system (see Section 2.1).

(1) *Categories* for organizing tag assignments.
(2) *Spatial information* (areas) describing to which part of a resource a tag assignment belongs to.
(3) *DBpedia URIs* that describe the semantic meaning of a tag assignment.

From the TagMe! data it seems that users appreciate those tagging facets, e.g. 899 of the 1264 tag assignments, which were performed within the three weeks after the launch of the system, were categorized and 657 times the users assigned a tag to a specific area within a picture. To better understand the context types available in TagMe!, we first present the results of preliminary analyses in which we investigated potential benefits of the categories and areas, and examined strategies for mapping tags to DBpedia URIs (see Section 5.1). In Section 5.2 we then summarize the results of our experiments that reveal the positive impact of the TagMe! context folksonomy on search.

5.1 Preliminary Analysis

In our preliminary analysis we first try to gain first insights into the characteristics of the three TagMe! context types. We target the following questions.

(1) How are *categories* used in comparison to tags and what are the benefits of categorizing tag assignments?
(2) What are the benefits of assigning tags to specific *areas* within an image (spatial information)?
(3) How accurately can tags (and categories) be mapped to *DBpedia URIs* describing the meaning of the annotations?

5.1.1 Analysis of Category Usage and Benefits

Figure 11 shows the evolution of the number of distinct tags and categories: Although categories can be entered freely like tags, they grow much less than tags. Further, only 31 of the 87 distinct categories (e.g., "car" or "sea") have also been used as tags, which means that users seem to use different kinds of concepts for categories and tags respectively.

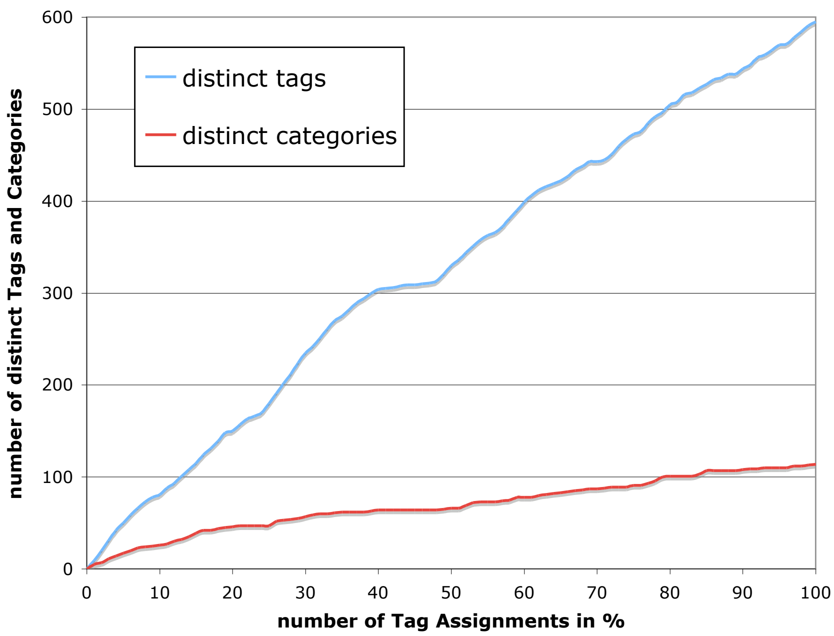


Figure 11. Growth of number of distinct tags in comparison to distinct categories.

The TagMe! system supports users in assigning categories by means of auto-completion (see Figure 1). During our evaluation we divided the users into two groups: 50% of the users (*group A*) got only those categories as suggestion, which they themselves used before, while the other 50% of the users (*group B*) got categories as suggestions, which were created by themselves or by another user within their group. This small difference in the functionality had a big impact on the alignment of the categories. The number of distinct categories in group A was growing 61.94% stronger than in group B. Hence, the vocabulary of the categories

Table 2. Identifying tags related to "clouds".

Rank	**Tag-based**	**Category-based**	**Area-based**
1	horse	sky	sky
2	sky	field	sun
3	tower	river	cloud
4	field	snow	cross
5	trees	water	sunset

can be aligned much better if categories, which have been applied by other users, are provided as suggestions as well.

Categories also enable to identify similar and related tags, which can, for example, be used for tag recommendations or query expansion. The identification of related tags is often based on tag co-occurrence analysis, e.g. (Sigurbjörnsson and van Zwol 2008), i.e. two tags are related if they are often assigned to the same resource.

Table 2 lists tags related to the tag "clouds". Here, the *tag-based* co-occurrence strategy does not perform that well as it also ranks tags such as "horse" or "field" within as the top five most related tags. The *category-based* strategy promotes basically those tags to the top of the ranking that share the most categories with "clouds". For example, "sky" and "clouds" share categories such as "nature" or "landscape". In general, the category-based strategy for detecting related tags seems to work better. However, in the given example, it still ranks the rather unrelated tag "field" very high. In our experiments, the best results are produced by the *area-based* strategy, which refines the category-based approach: It ranks those tags higher that occur in spatial tag assignments, whose areas overlap with the areas of the given tag. As shown in Table 2, it also produces—in comparison to the other strategies—the most reasonable ranking of tags related to "clouds". Four of the top five tags are apparently related ("cross" seems to be the only exception).

From our initial experiments on identifying similar tags, we draw the conclusion that tags, which share the same category and are often assigned to similar areas within an image (cf. *Area-based*), are closer related than tags that often co-occur at same resource.

5.1.2 Analysis of Spatial Tagging Information

Categories can be differentiated according to their usage. For example, some categories have never or very seldomly been used when a specific area of an image was tagged (e.g., "time", "location", or "art") while others have been applied almost only for tagging a specific area (e.g., "people", "animals", or "things").

The areas, can moreover be used to learn relations among categories and tags. Figure 12 shows (i) the areas that have been annotated whenever the categories "people" and "friends" have been used (the darker an area the more tags have been assigned to that area). As the areas that have been tagged in both categories strongly correlate and as category "people" was used more often than category "friends" one can deduce that "friends" is possibly a *sub-category* of "people" even if both categories would never co-occur at the same resource. Relations between tags can also deduced by analyzing the tagged areas. Figure 12 shows (ii) the

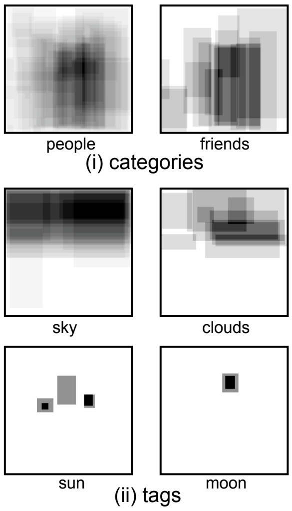


Figure 12. Annotated areas.

areas that were tagged with “sky”, “clouds”, “sun”, and “moon”[1] and via the size and position of the area it is possible to learn that an entity *is part of* or *contained in* another entity (e.g., “sun, moon, and clouds are contained in sky”). The learned relations among tags and categories can moreover be used to learn and refine relations between URIs (ontology concepts) as TagMe! maps tags and categories to DBpedia URIs.

5.1.3 Mapping to DBpedia URIs

For realizing the feature of mapping tags and categories to DBpedia (Auer *et al.* 2007) URIs we compared the following two strategies.

- **DBpedia Lookup** The naive lookup strategy invokes the DBpedia lookup service with the tag/category that should be mapped to a URI as search query. DBpedia ranks the returned URIs similarly to PageRank (Bizer *et al.* 2009) and our naive mapping strategy simply assigns the top-ranked URI to the tag/category in order to define its meaning.
- **DBpedia Lookup + Feedback** The advanced mapping strategy is able to consider feedback while selecting an appropriate DBpedia URI. Whenever a tag/category is assigned, for which already a correctly validated DBpedia URI exists in the TagMe! database then that URI is selected. Otherwise the strategy falls back the naive DBpedia Lookup.

Figure 13 shows the accuracy of both strategies. The mappings of the naive approach result in a precision of 79.92% for mapping tags to DBpedia URIs and 84.94% for mapping categories considering only those tag assignments where a DBpedia URI that describes the meaning properly exists. The consideration of feedback improves the precisions of the naive DBpedia Lookup clearly to 86.85% and 93.77% respectively, which corresponds to an improvement of 8.7% and 10.4%. As the mapping accuracy for categories is higher than the one for tags, it seems that the identification of meaningful URIs for categories is easier than for tags. In summary, the results of the DBpedia mapping are very encouraging. Moreover, the precision of the category mappings, which are determined by the strategy that incorporates feedback, will further improve, because—fostered by TagMe!’s

[1]The visualizations are based on 25 (“sky”), 10 (“clouds”), 6 (“sun”), and 2 (“moon”) tag assignments respectively.

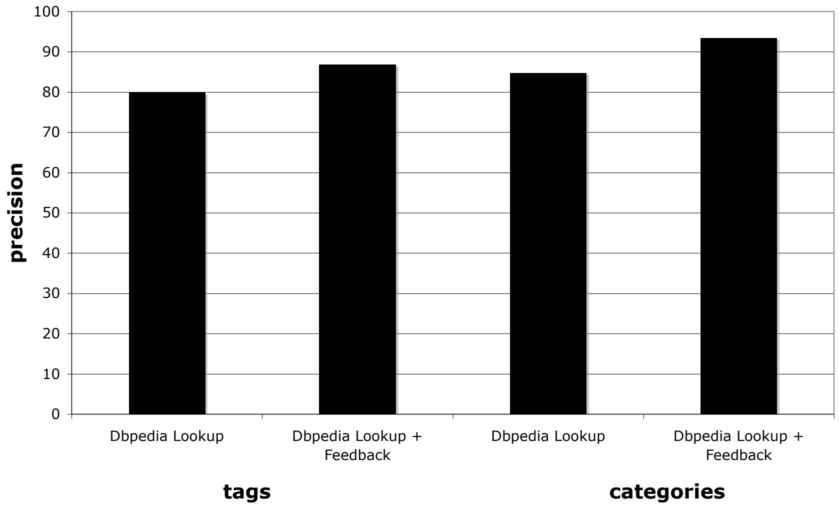


Figure 13. Precision of mapping tags and categories to DBpedia URI.

category suggestion feature—the number of distinct categories seems to converge (cf. Figure 11). Further, the mapping strategies itself can be enhanced by also considering the context of the tag/category that should be mapped. For example, for mapping a tag assignment one could select the DBpedia URI, which best fits to the DBpedia URI of the category that is associated to the tag assignment. The implementation of such advanced mapping strategies is part of our future work.

The DBpedia mapping reduces the number of distinct tags and categories within TagMe! by 14.1% and 20.9% respectively, which promises a positive impact on the recall when executing tag-based search. For example, while some users assigned the tag “car” to pictures showing cars other users chose “auto” to annotate other pictures that show cars. As both kinds of tag assignments are mapped to “http://dbpedia.org/resource/Automobile”, TagMe! can simply search via the DBpedia URI whenever users search via “car” or “auto” to increase recall of the tag-based search operations.

5.1.4 Synopsis

In summary, the context types available in the TagMe! folksonomy have a positive impact on identifying correlations between the folksonomy entities (e.g. identifying similar tags). Further, categories and areas enable the extraction of additional semantic relations between tags. As tags are mapped to DBpedia URIs that describe the meaning of a tag assignment, it is possible to deduce rich semantics from the context folksonomy available in TagMe!. The results of our preliminary analysis can be summarized as follows.

- The usage of categories differs from the usage of tags: Even for those users, who did not benefit from the category suggestions, the number of distinct categories is growing slower than the number of distinct tags.
- For identifying related tags, tag assignments enriched with category and area context seem to be a more valuable source of information than traditional tag assignments: Tags, which share the same categories and are often assigned to similar areas within an image, are closer related than tags that simply co-occur at same resources.
- The spatial tag assignments can be used to learn typed relations among tags and categories such as *sub-category*, *sub-tag*, *part-of*, or *contained-in* relations. As tags and category assignments are mapped to meaningful URIs (ontological concepts), it is possible to propagate the learned relations to ontologies.

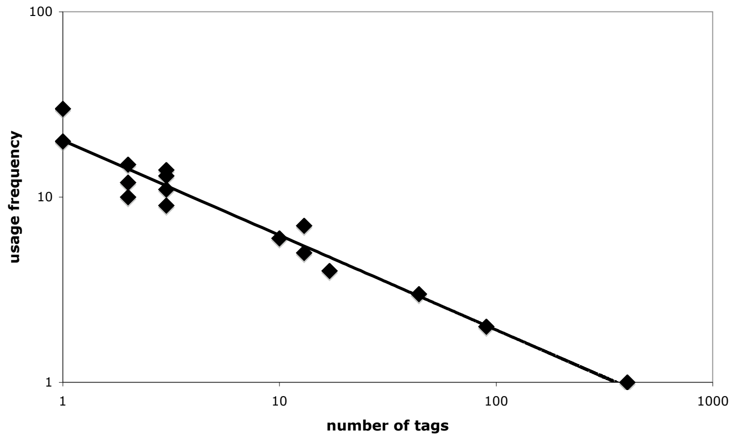


Figure 14. Tag usage in the TagMe! data set on a logarithmic scale. Only a few distinct tags have been used frequently while most of the tags are only used once.

- A naive DBpedia lookup allows us to map tags and categories to ontological concepts (DBpedia URIs) with a high precision of 79.92% (tags) and 84.94% (categories). The consideration of feedback improves the accuracy of the mapping of tags and categories to 86.85% and 93.77% respectively.

This preliminary analysis already delivers insights into the potential of the context information available in the TagMe! folksonomy. In the section we will evaluate whether categories, the size and position of areas, and the DBpedia URI assignments can be applied to improve the quality of search and ranking.

5.2 Search Evaluation

In our search evaluation we examine the impact of the advanced semantics provided by the TagMe! context folksonomy on search. In particular, we apply the FolkRank algorithm (see Section 3.1) as well as the Category-, Area-, and URI-based FolkRank adaptions to search and rank Flickr images and investigate how the different context types can help to improve the search performance. Similarly to the evaluation outlined in Section 4 we evaluate the algorithms with respect to the following task.

> **Ranking Task.** *Given a keyword query (tag), the task of the ranking strategy is to compute a ranking of resources so that resources that are most relevant to the keyword query appear at the top of the ranking.*

Our primary goal is to determine whether the additional context information has a positive impact on the ranking task. We further examine the characteristic strengths and weaknesses of the different context types by comparing the ranking performance of the corresponding FolkRank-based strategies.

5.2.1 DataSet and Test Set

We conducted our experiments on the TagMe! data set that evolved during the first month after the launch of the system. In this period the users created 1264 tag assignments where 899 tag assignments were also enriched with a category and 657 tag assignments were attached to a specific area of a Flickr resource. As outlined in Section 5.1.1, the number of distinct tags was growing faster than the number of distinct categories. Finally, the TagMe! data set contained 610 distinct tags and 118 distinct categories. The distribution of the usage frequency of tags (see Figure 14) shows the same characteristics as detected in the data set used in Section 4. While some tags are used very often, the majority of tags are used just once.

The DBpedia URI assignments that were automatically attached by TagMe! were validated by hand so that the data set on which we performed the experiments did not contain wrong URI assignments. The cleaned data set finally contained 360 distinct DBpedia URIs referenced by tags and 92 DBpedia URIs referenced by categories. For 17% of the tag assignments there did not exist a correct DBpedia URI mappings.

The relevance assessment was done similarly to the procedure described in Section 4.2. We selected 24 representative tags (according to the usage frequency, cf. Figure 14) as keyword queries and asked TagMe! users to rate the relevance of a picture to a given query on a five-point scale: *yes*, *rather yes*, *rather no*, *no*, and *don't know*. Therefore, for each of the queries we obtained all the relevant resources in the TagMe! data set. On average, for each query there were nearly 30 resources in the data set that were rated as relevant (*yes*). However, four of the queries had below 10 relevant (*yes*) resources. For all the 24 tag-based queries a proper DBpedia URI was available in the data set.

5.2.2 Method and Metrics

The *ranking task* defined at the beginning of the section requires the strategies to arrange those resources at the top of the ranking that are most relevant to the given query. We analyzed the ranking algorithms presented in Section 3 that are applicable to the TagMe! context folksonomy: FolkRank, Category-based FolkRank (CategoryFolkRank), Area-based FolkRank (AreaFolkRank), and URI-based FolkRank (DBpediaFolkRank). Each ranking strategy then had to compute a resource ranking for each of the 24 representative keyword queries. We measured the quality of the rankings using the precision at rank k ($P@K$), which represents the average proportion of *relevant* items within the top k (cf. Section 4.3). For our experiment we considered an item as *relevant* iff the average user judgement is at least "yes".

In addition to the FolkRank-based approaches we also consider a ranking algorithm (denoted as "F+C+A+D") that combines all four ranking strategies: Given the list of weighted resources as computed by the different algorithms it utilizes the average ranking weight to rank the resources.

Following our experiments presented in Section 4, we tested the statistical significance of our results with a two-tailed t-Test with a significance level of $\alpha = 0.05$. The null hypothesis H0 is that some strategy s_1 is as good as another strategy s_2, while H1 states that s_1 is better than s_2.

5.2.3 Results

Figure 15 shows the precisions (P@10 and P@20) of the different ranking strategies. Those algorithms that make use of context information embedded in the folksonomy perform better than the traditional FolkRank algorithm that considers only the tag assignments without any additional context. Between DBpediaFolkRank and FolkRank there seems to be no remarkable performance difference. However, as noted in Section 5.2.1, the DBpediaFolkRank is operating on 215 fewer tag assignments than the other algorithms. It is thus remarkable that DBpediaFolkRank still performs slightly better than FolkRank. The CategoryFolkRank presents good results especially with respect to the precision within the top 20 (P@20). Hence, the hypothesis raised in Section 3.2.3 seems to hold: category assignments can be used to relate resources. By exploiting the category context, the algorithm detects relevant resources that are not directly related via tag assignments to the given query. The AreaFolkRank algorithm, which exploits the size and position of spatial information attached to the tag assignments, is—with respect to P@10—the best algorithm among the core ranking strategies (P@10 = 52.9%). However, there is no significant difference between the FolkRank and the Area-, Category-, and DBpedia-based FolkRank.

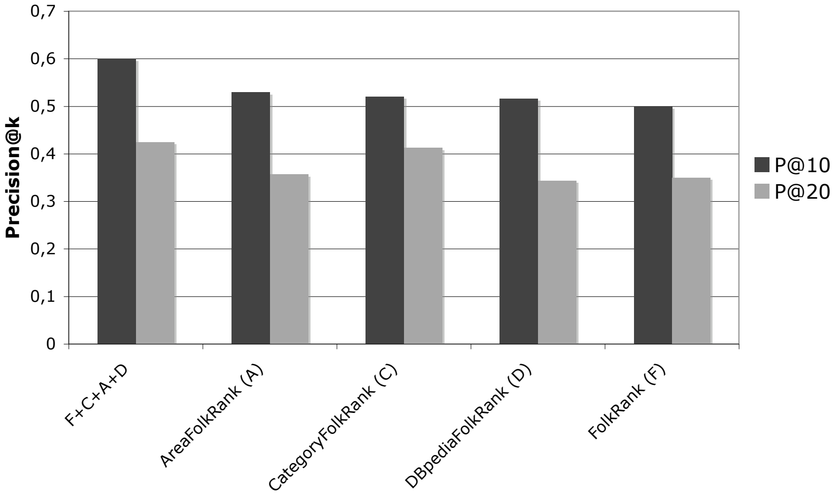


Figure 15. Precisions of FolkRank-based search algorithms.

The strategy "F+C+A+D", which combines all four core ranking strategies (i.e., FolkRank, CategoryFolkRank, AreaFolkRank, and DBpediaFolkRank), is the most successful strategy. It performs significantly better than the FolkRank algorithm regarding the P@10 and P@20 metrics. The combined strategy improves the precision of FolkRank by 20.0% and 21.4% with respect to the precision within the top 10 and top 20 respectively.

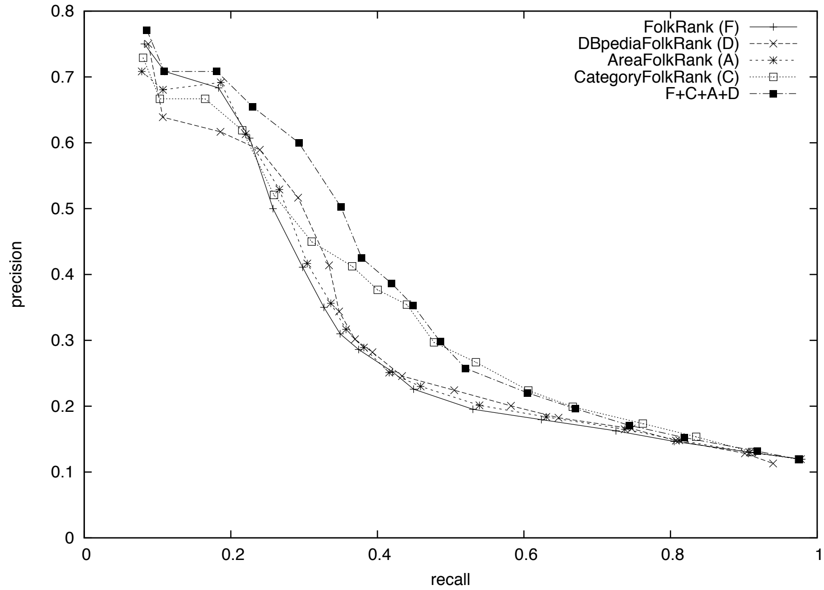


Figure 16. Precision recall diagram of FolkRank-based search algorithms.

Figure 16 depicts the precision-recall diagram of the different FolkRank-based ranking algorithms. It underlines that the context-based approach, which combines FolkRank with the strategies that exploit the category, area, and DBpedia context, is the best performing ranking strategy as it results in the best precision-recall ratio. In the low recall interval, i.e. within the very top of the resource rankings, FolkRank can compete with the other algorithms. For example, the probability that a relevant resource appears at the first rank is 75.0% for FolkRank and 79.2% for the combined strategy. However, with higher recall values, the precision of FolkRank drops significantly stronger than the one of the Category-based FolkRank or the combined strategy F+C+A+D: At a recall level of 0.5 the precision of F+C+A+D and CategoryFolkRank is 0.29 and therewith significantly higher (approx. 45%) as the precision of FolkRank.

In summary, the exploitation of context embedded in the folksonomy is beneficial for ranking resources. While the size and position of the area helps to improve the precision particularly at the top of the resource rankings, the DBpedia and category context successfully contribute to improve the recall. And by combining the different context types we are able to improve the ranking performance of FolkRank significantly.

6. Conclusions and Future Work

Ranking in folksonomies is currently an important research topic. In this article we proposed different approaches for improving ranking approaches by exploiting context information. We introduced a model for integrating context information into folksonomies and presented an approach that allows the adaption of rankings to the actual context of a user independently from the underlying ranking algorithm. We presented different strategies that are able to construct context from user interactions (clicks) by the notion of tag clouds even if no previous knowledge about the user is available. Furthermore, we introduced FolkRank-based algorithms for exploiting context information and SocialHITS, a new ranking algorithm for folksonomy systems. We showed that SocialHITS significantly improves the HITS-based approach proposed in (Wu *et al.* 2006).

We analyzed the performance of SocialHITS and other folksonomy-based ranking algorithms for the task of contextualizing rankings while considering different types of context and revealed that those strategies, which do not require any previous knowledge about the user, perform significantly better than tag-based user modeling. For example, by considering the the tag cloud of a resource the user has just visited we are able to adapt the ranking of a subsequent search activity to the user's current context. A remarkable feature of our evaluation was that we also measured the ranking performance with respect to the task of ranking users, which is new in the field of research on folksonomies and further promises high impact on the future of social networking. Here, we identified SocialHITS as one of the most promising ranking algorithms. Our evaluations further reveal that the exploitation of context embedded in folksonomies, e.g. categories and URIs attached to individual tag assignments, are beneficial to learning relationships between tags and help to improve search significantly. In addition to our extensive evaluations we showcase our approaches in the TagMe! system, a tagging and exploration interface for Flickr pictures.

References

Abel, F., Baldoni, M., Baroglio, C., Henze, N., Krause, D. and Patti, V., 2009a. Context-based ranking in folksonomies. *In*: C. Cattuto, G. Ruffo and F. Menczer, eds. *Hypertext* ACM, 209–218.

Abel, F., Frank, M., Henze, N., Krause, D., Plappert, D. and Siehndel, P., 2007. GroupMe! – Where Semantic Web meets Web 2.0. *In*: Aberer et al., ed. *The Se-*

mantic Web, 6th International Semantic Web Conference (ISWC), 2nd Asian Semantic Web Conference (ASWC), Busan, Korea. Springer Verlag.

Abel, F., Henze, N. and Krause, D., 2008. Exploiting additional Context for Graph-based Tag Recommendations in Folksonomy Systems. *In*: *Int. Conf. on Web Intelligence and Intelligent Agent Technology*, Sydney, Australia. ACM Press.

Abel, F., Henze, N., Krause, D. and Kriesell, M., 2009b. On the Effect of Group Structures on Ranking Strategies in Folksonomies. *In*: R. Baeza-Yates and I. King, eds. *Weaving Services and People on the World Wide Web*, July. Springer Verlag, 275–300.

Abel, F., Kawase, R., Krause, D. and Siehndel, P., 2009c. Multi-faceted Tagging in TagMe!. *In*: *8th International Semantic Web Conference (ISWC2009)*, Washington, DC, USA. Springer Verlag.

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R. and Ives, Z., 2007. DBpedia: A Nucleus for a Web of Open Data. *In*: Aberer et al., ed. *The Semantic Web, 6th International Semantic Web Conference (ISWC), 2nd Asian Semantic Web Conference (ASWC)*, Busan, Korea. Springer Verlag, 715–728.

Bao, S., Xue, G., Wu, X., Yu, Y., Fei, B. and Su, Z., 2007. Optimizing Web Search using Social Annotations. *In*: *Proc. of 16th Int. World Wide Web Conference (WWW '07)*, Banff, Canada. ACM Press, 501–510.

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R. and Hellmann, S., 2009. DBpedia - A crystallization point for the Web of Data. *Web Semantics: Science, Services and Agents on the World Wide Web.*

Byde, A., Wan, H. and Cayzer, S., 2007. Personalized Tag Recommendations via Tagging and Content-based Similarity Metrics. *In*: *Proc. of the Int. Conf. on Weblogs and Social Media (ICWSM)*, Washington, DC, USA.

Chirita, P.A., Firan, C. and Nejdl, W., 2007. Personalized Query Expansion for the Web. *In*: *Proceedings of the 30th Int. ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR '07)*, Amsterdam, The Netherlands. ACM Press, 7–14.

Dellschaft, K. and Staab, S., 2008. An epistemic dynamic model for tagging systems. *In*: P. Brusilovsky and H.C. Davis, eds. *Hypertext.* ACM Press, 71–80.

Firan, C.S., Nejdl, W. and Paiu, R., 2007. The Benefit of Using Tag-Based Profiles. *In*: *Proc. of the 2007 Latin American Web Conference (LA-WEB 2007)*, Washington, DC, USA. IEEE Computer Society, 32–41.

Halpin, H., Robu, V. and Shepherd, H., 2007. The Complex Dynamics of Collaborative Tagging. *In*: *Proc. of 16th Int. World Wide Web Conference (WWW '07)*, Banff, Canada. ACM Press, 211–220.

Hotho, A., Jäschke, R., Schmitz, C. and Stumme, G., 2006a. BibSonomy: A Social Bookmark and Publication Sharing System. *In*: *Proc. First Conceptual Structures Tool Interoperability Workshop*, Aalborg, Denmark. 87–102.

Hotho, A., Jäschke, R., Schmitz, C. and Stumme, G., 2006b. Emergent Semantics in BibSonomy. *In*: C. Hochberger and R. Liskowsky, eds. *Informatik 2006: Informatik für Menschen*, Vol. 94(2) of *LNI.* Bonn, Germany: GI.

Hotho, A., Jäschke, R., Schmitz, C. and Stumme, G., 2006c. Information Retrieval in Folksonomies: Search and Ranking. *In*: *Proceedings of the 3rd European Semantic Web Conference*, Vol. 4011 of *LNCS*, Budva, Montenegro. Springer Verlag, 411–426.

Jäschke, R., Marinho, L.B., Hotho, A., Schmidt-Thieme, L. and Stumme, G., 2007. Tag Recommendations in Folksonomies. *In*: *Proc. 11th Europ. Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, 506–514.

Kim, M.C. and Choi, K.S., 1999. A Comparison of Collocation-based Similarity Measures in Query Expansion. *Information Processing and Management*, 35

(1), 19–30.

Kleinberg, J.M., 1999. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46 (5), 604–632.

Li, X., Guo, L. and Zhao, Y.E., 2008. Tag-based Social Interest Discovery. *In*: *Proc. of the 17th Int. World Wide Web Conference (WWW'08)*, Beijing, China. ACM Press, 675–684.

Marlow, C., Naaman, M., Boyd, D. and Davis, M., 2006a. HT06, tagging paper, taxonomy, Flickr, academic article, to read. *In*: *Proc. of the 17th Conf. on Hypertext and Hypermedia*, ACM Press, 31–40.

Marlow, C., Naaman, M., Boyd, D. and Davis, M., 2006b. Position Paper, Tagging, Taxonomy, Flickr, Article, ToRead. *In*: *Collaborative Web Tagging Workshop at WWW '06*, Edinburgh, Scotland. ACM Press.

Michlmayr, E. and Cayzer, S., 2007. Learning User Profiles from Tagging Data and Leveraging them for Personal(ized) Information Access. *In*: *Proc. of the Workshop on Tagging and Metadata for Social Information Organization, 16th Int. World Wide Web Conference (WWW '07)*, Banff, Canada. ACM Press.

Mika, P., 2005. Ontologies Are Us: A unified model of social networks and semantics. *In*: *Proc. Int. Semantic Web Conference (ISWC 2005)*, Galway, Ireland. Springer Verlag, 522–536.

Nauerz, A. and Groh, G., 2008. Implicit Social Network Constructuon and Expert User Determination in Web Portals. *In*: *AAAI Spring Symposium on Social Information Processing*. AAAI Press, 60–65.

Page, L., Brin, S., Motwani, R. and Winograd, T., The PageRank Citation Ranking: Bringing Order to the Web. Technical report, 1998, Stanford Digital Library Technologies Project.

Passant, A. and Laublet, P., 2008. Meaning Of A Tag: A collaborative approach to bridge the gap between tagging and Linked Data. *In*: *Proceedings of the WWW 2008 Workshop Linked Data on the Web (LDOW2008), Beijing, China*.

Rattenbury, T., Good, N. and Naaman, M., 2007. Towards automatic extraction of event and place semantics from flickr tags. *In*: *Proc. of the 30th Int. ACM SIGIR Conf. on Information Retrieval (SIRIR '07)*. ACM Press, 103–110.

Russell, B.C., Torralba, A.B., Murphy, K.P. and Freeman, W.T., 2008. LabelMe: A Database and Web-Based Tool for Image Annotation.. *International Journal of Computer Vision*, 77 (1-3), 157–173.

Sigurbjörnsson, B. and van Zwol, R., 2008. Flickr Tag Recommendation based on Collective Knowledge. *In*: *Proc. of 17th Int. World Wide Web Conference (WWW '08)*, Beijing, China. ACM Press, 327–336.

Vander Wal, T., 2007. Folksonomy. *http://vanderwal.net/folksonomy.html*.

Voorhees, E.M., 1994. Query expansion using lexical-semantic relations. *In*: *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, Dublin, Ireland. Springer Verlag, 61–69.

Wu, H., Zubair, M. and Maly, K., 2006. Harvesting social knowledge from folksonomies. *In*: *Proceedings of the seventeenth conference on Hypertext and hypermedia (HYPERTEXT '06)*. ACM Press, 111–114.