CONTEXT-AWARE RANKING ALGORITHMS IN FOLKSONOMIES

Fabian Abel, Nicola Henze and Daniel Krause
IVS – Semantic Web Group, Leibniz University Hannover, Appelstr. 4, 30167 Hannover, Germany

Keywords: Social Media, Search, Ranking, Folksonomies, GroupMe!

Abstract: Folksonomy systems have shown to contribute to the quality of Web search ranking strategies. In this paper, we analyze and compare graph-based ranking algorithms: FolkRank and SocialPageRank. We enhance these algorithms by exploiting the context of tags, and evaluate the results on the GroupMe! dataset. In GroupMe!, users can organize and maintain arbitrary Web resources in self-defined groups. When users annotate resources in GroupMe!, this can be interpreted in context of a certain group. The grouping activity itself is easy for users to perform: simple drag-and-drop operations allow users to collect and group resources. However, it delivers valuable semantic information about resources and their context. We show how to use this information to improve the detection of relevant search results, and compare different strategies for ranking result lists in folksonomy systems.

1 INTRODUCTION

Social interactions, participation in the content creation process, easy-to-use applications – these are among the usage characteristics of currently successful, so-called Web 2.0-applications. Users in Web 2.0 applications are more than ever active in the content life-cycle: They contribute with their opinion by annotating Web content (the so-called *tagging*), they add and annotate content (e.g. by using applications for sharing their bookmarks, pictures, videos, etc. with other users), they rate content, and they create content (e.g. with sorts of online dictionaries, so-called *blogs*). In this paper, we focus on the first type of applications: social tagging systems. In a social tagging system, users tag Web content, share these tags with other users of the application, and profit by the tagging activity of the whole user community by discovering / retrieving relevant Web content during browsing / as answers to search queries. The tagging activities are modeled in a *folksonomy* (Vander Wal, 2007): a taxonomy, which evolves over time when users (the *folks*) annotate resources with freely chosen keywords. Folksonomies can be divided into *broad* folksonomies, which allow different users to assign the same tag to the same resource, and *narrow* folksonomies, in which the same tag can be assigned to a resource only once.

Bao et al. showed that Web search can be improved by exploiting knowledge embodied in folksonomies (Bao et al., 2007). In this paper, we introduce and evaluate different ranking strategies for folksonomy systems. In particular, we

- propose an algorithm, which exploits the context gained by grouping resources in folksonomy systems and which improves search for resources: GRank.
- compare existing ranking algorithms for folksonomies: FolkRank and SocialPageRank. We extend these algorithms and propose (a) group-sensitive FolkRank algorithms, and (b) a topic-sensitive SocialPageRank algorithm, and evaluate their quality with respect to search tasks.

The paper is organized as follows. In the next section we discuss our work with respect to related work. In Section 3, we briefly introduce the functionality of the GroupMe! system. Section 4 presents a formal definition of folksonomies, and their extension with group structures. Afterwards, we identify the characteristics of the folksonomy, which builds the dataset of the GroupMe! application. Different folksonomy-based ranking strategies are discussed in the following section. Section 6 presents our evaluation results. The paper ends with conclusions.

2 RELATED WORK

In this paper we enhance ranking algorithms for folksonomies. We extend the FolkRank algorithm introduced in (Hotho et al., 2006b) with the capability of

Abel F., Henze N. and Krause D.
CONTEXT-AWARE RANKING ALGORITHMS IN FOLKSONOMIES.
DOI: 10.5220/0001838601670174
In *Proceedings of the Fifth International Conference on Web Information Systems and Technologies (WEBIST 2009)*, page
ISBN: 978-989-8111-81-4

exploiting additional context information gained by GroupMe! folksonomies. Furthermore, we improve SocialPageRank (Bao et al., 2007) by enabling *topic-sensitive* rankings. In our experiment we evaluate ranking of resources whereas in (Abel et al., 2008a) we focussed on ranking tags and evaluated the quality of different graph-based ranking algorithms with respect to tag recommendations. In (Sigurbjörnsson and van Zwol, 2008) the authors propose an approach for recommending tags, which is based on co-occurences of tags. However, our evaluations in (Abel et al., 2008a) indicate that graph-based recommender algorithms are more appropriate for folksonomies than strategies as described in (Sigurbjörnsson and van Zwol, 2008).

When designing algorithms for folksonomy systems, the basic assumption is that tags describe the content of resources very well. In (Li et al., 2008) the authors prove this assumption by comparing the actual content of web pages with tags assigned to these websites in the del.icio.us[1] system.

3 GROUPME! FOLKSONOMY SYSTEM

The GroupMe![2] Folksonomy System (Abel et al., 2007) is a fun-to-use Web 2.0 application. It is a resource sharing system like del.icio.us or Bibsonomy[3], offering the extended feature of grouping Web resources. These GroupMe! groups can contain arbitrary multimedia resources like websites, photos or videos, which are visualized according to their media type: E.g., images are displayed as thumbnails and the headlines from RSS feeds are structured in a way that the most recent information are accessible by just one click. With this convenient visualization strategy, the user can grasp the content immediately without the need of visiting the original Web resource. Figure 1 shows a group about WEBIST 2009 in Lisbon, which contains the website of the conference, a video with traveling information about Lisbon, a GroupMe! group about the last WEBIST conference, etc.

GroupMe! groups are created by dragging & dropping multimedia resources from various sources into a group (cf. Figure 1). We also offer a *bookmarklet* to add the currently visited Web site with a single click into the GroupMe! system. Building groups is a very convenient way of aggregating content. As groups are also normal Web resources, it is

[1]http://del.icio.us
[2]http://groupme.org
[3]http://bibsonomy.org

Figure 1: GroupMe! group about WEBIST ’09 available at http://groupme.org/GroupMe/group/2671.

possible to group groups and hence, to model hierarchies.

To foster Semantic Mashups, all data collected by the GroupMe! system is available in different formats: For a lightweight integration, we offer RSS feeds, which enable users easily to get informed if new content is added to a group. Furthermore, we offer an RDF based RESTful API, which enables other applications to navigate through the semantically enriched GroupMe! data corpus according to the principles of Linked Data.

4 FOLKSONOMIES

Formally, folksonomies are defined as tuples of folksonomy entities, i.e. users, tags, and resources, and the bindings between these entities, which are called *tag assignments* and denote which user has assigned which tag to a certain resource. According to (Hotho et al., 2006a), a folksonomy can be defined as

Definition 1 (Folksonomy). 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, $Y \subseteq U \times T \times R$

GroupMe! extends this folksonomy definition by the concept of groups:

Definition 2 (Group). A *group* is a finite set of resources.

A group is a resource as well. Groups can be tagged or arranged in groups, which effects hierarchies among resources. In general, tagging of resources within the GroupMe! system is done in context of a group. Hence a GroupMe! folksonomy is formally characterized via Definition 3 (cf. (Abel et al., 2007)).

Definition 3 (GroupMe! folksonomy). A GroupMe! 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
- $\breve{R} = R \cup G$ is the union of the set of resources and the set of groups
- $\breve{Y}$ defines a *GroupMe! tag assignment*: $\breve{Y} \subseteq U \times T \times \breve{R} \times (G \cup \{\varepsilon\})$, where ε is a reserved symbol for the *empty group context*, i.e. a group that is not contained in another group when it is tagged by a user

4.1 Folksonomy Characteristics in GroupMe!

To decide whether known folksonomy search and ranking algorithms can be improved by considering the group context, we had a closer look on the tagging and grouping behavior of our users by analyzing a snapshot of the GroupMe! dataset, which contains 1546 unique tags, 2338 resources, 352 users, 453 groups, and 2690 tag assignments. The first question was whether users made use of the feature of grouping and visualizing different media types. In Figure 2 we show the distribution of the different media types in the GroupMe! system.

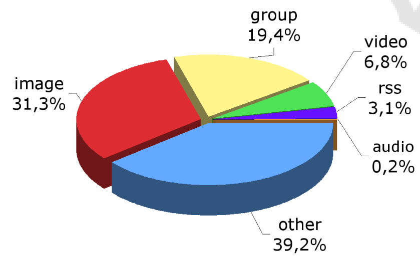

Figure 2: Media type distribution in the GroupMe! system.

Our observation is, that users use different mediatypes and especially multimedia documents. About 40% of all resources in our system are multimedia documents, where tags form the main textual description, because extraction of other meta data is barely possible.

In (Li et al., 2008) the authors show that tags describe resources very precisely and are hence a valuable input for searching and ranking. GroupMe! motivates users to tag resources by using the free-for-all tagging approach (see (Marlow et al., 2006)), which enables users to tag not only their own resources, but all resources within the GroupMe! system.

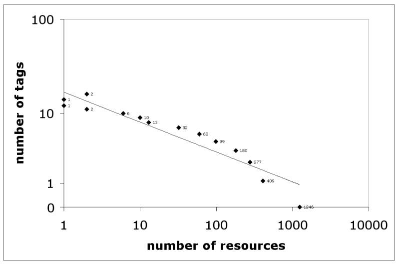

Figure 3: Distribution of tag assignments.

On a logarithmic scale (extended with zero), we plotted the number of tag assignments on the y-axis and the number of resources having this number of tags assigned on the x-axis (see Figure 3). We observed a power law distribution of the tag assignments per resource, while about 50% of all resources do not even have a single tag assignment. That means, that 50% of all resources in the GroupMe! system can hardly be found by known folksonomy based search and ranking algorithms.

5 FOLKSONOMY-BASED RANKING ALGORITHMS

In this section we present different algorithms, which target on ranking folksonomy entities. We first introduce graph-based algorithms that can be applied to arbitrary folksonomy entities (users, tags, and resources). In Section 5.2 we describe algorithms, which specifically focus on ranking resources to support e.g. traditional search functionality in folksonomy systems.

Our contributions, i.e. ranking algorithms we developed, can be summarized as follows:

GFolkRank & GFolkRank^{+} . Graph-based ranking algorithms, which extend FolkRank (Hotho et al., 2006b) and turn it into a group-sensitive algorithm in order to exploit GroupMe! folksonomies (see Section 5.1.2).

Personalized SocialPageRank . Extension to SocialPageRank (Bao et al., 2007), which allows for topic-sensitive rankings.

GRank . A search and ranking algorithm optimized for GroupMe! folksonomies.

5.1 Universal Ranking Strategies

Universal ranking strategies like FolkRan and Group-sentitive FolkRank can be used to rank arbitrary parts of a folksonomy, e.g. users, resources, tags etc.

5.1.1 FolkRank

FolkRank (Hotho et al., 2006b) adapts Personalized PageRank (Page et al., 1998) for ranking users, tags, and resources in traditional folksonomies.

$$\vec{w} \leftarrow dA\vec{w} + (1-d)\vec{p} \tag{1}$$

The adjacency matrix A models the folksonomy graph $\mathbb{G}_\mathbb{F} = (V_\mathbb{F}, E_\mathbb{F})$. $\mathbb{G}_\mathbb{F}$ is an undirected, weighted tripartite graph, which is created from the the folksonomy (cf. Definition 1). The set of nodes is $V_\mathbb{F} = U \cup T \cup R$ and the set of edges is given via $E_\mathbb{F} = \{\{u,t\},\{t,r\},\{u,r\} | (u,\, t, r) \in Y\}\}$. The edges are weighted according to their frequency within the set of tag assignments. For example, $w(t,r) = |\{u \in U : (u,t,r) \in Y\}|$ denotes the popularity of tag t for the resource r and counts the number of users, who have annotated r with t. $w(u,t)$ and $w(u,r)$ are defined accordingly. A is normalized so that each row has a 1-norm equal to 1. The influence of the preference vector $\vec{p}$ is configured via $d \in [0,1]$. Finally, the FolkRank algorithm is defined as follows (see (Hotho et al., 2006b)).

Definition 4 (FolkRank). The *FolkRank algorithm* computes a topic-specific ranking in folksonomies by executing the following steps:

1. $\vec{p}$ specifies the preference in a topic (e.g. preference for a given tag).
2. $\vec{w}_0$ is the result of applying the Personalized PageRank with $d = 1$.
3. $\vec{w}_1$ is the result of applying the Personalized PageRank with some $d < 1$.
4. $\vec{w} = \vec{w}_1 - \vec{w}_0$ is the final weight vector. $\vec{w}[x]$ denotes the *FolkRank* of some $x \in V_\mathbb{F}$.

When applying FolkRank to GroupMe! folksonomies (see Definition 3) a straightforward approach is to ignore the group dimension of GroupMe! tag assignments. Therewith, the construction of the folksonomy graph $\mathbb{G}_\mathbb{F} = (V_\mathbb{F}, E_\mathbb{F})$ has to be adapted slightly. The set of nodes is given by $V_\mathbb{F} = U \cup T \cup \check{R}$ and $E_\mathbb{F} = \{\{u,t\},\{t,r\},\{u,r\} | u \in U, t \in T, r \in \check{R}, g \in G \cup \{\varepsilon\}, (u,t,r,g) \in \check{Y}\}$ defines the set of edges. Computation of weights is done correspondingly to the FolkRank algorithm, e.g. $w(t,r) = |\{u \in U : g \in G \cup \{\varepsilon\}, (u,t,r,g) \in \check{Y}\}|$ is the number of users, who annotated resource r with tag t in any group.

5.1.2 Group-sensitive FolkRank (GFolkRank)

The traditional FolkRank does not make use of the additional structure of GroupMe! groups. In (Abel et al., 2008b) we propose different adaptations of FolkRank, which exploit group structures in folksonomies, and show that they improve the ranking quality of FolkRank significantly (one-tailed t-test, significance level $\alpha = 0.05$). In this paper we present one of these strategies, which we call *GFolkRank*.

GFolkRank interprets groups as artificial, unique tags. If a user u adds a resource r to a group g then GFolkRank interprets this as a tag assignment (u, t_g, r, ε), where $t_g \in T_G$ is the artificial tag that identifies the group. The folksonomy graph $\mathbb{G}_\mathbb{F}$ is extended with additional vertices and edges. The set of vertices is expanded with the set of artificial tags T_G: $V_\mathbb{G} = V_\mathbb{F} \cup T_G$. Furthermore, the set of edges $E_\mathbb{F}$ is augmented by $E_\mathbb{G} = E_\mathbb{F} \cup \{\{u,t_g\},\{t_g,r\},\{u,r\} | u \in U, t_g \in T_G, r \in \check{R}$, u has added r to group $g\}$. The new edges are weighted with a constant value w_c as a resource is usually added only once to a certain group. We select $w_c = 5.0 \approx max(|w(t,r)|)$ because we believe that grouping a resource is, in general, more valuable than tagging it. GFolkRank is consequently the FolkRank algorithm (cf. Section 5.1.1), which operates on basis of $\mathbb{G}_\mathbb{G} = (V_\mathbb{G}, E_\mathbb{G})$.

GFolkRank$^+$ denotes a strategy that extends GFolkRank with the feature of propagating tags, which have been assigned to a group, to its resources. The weight of edges $e \in E_\mathbb{G}$, which are caused by such inherited tag assignments, is adjusted by a dampen factor $df \in [0,1]$. For our evaluations in Section 6 we set $df = 0.2$.

5.2 Ranking Resources

In contrast to the FolkRank-based algorithms, which can be utilized to rank all types of folksonomy entities – i.e. users, tags, resources, and groups – we present SocialPageRank (Bao et al., 2007) and propose GRank, which is a search and ranking algorithm optimized for GroupMe! folksonomies. Both algorithms concentrate on ranking resources.

5.2.1 SocialPageRank

The SocialPageRank algorithm (Bao et al., 2007) is motivated by the observation that there is a strong interdependency between the popularity of users, tags, and resources within a folksonomy. For example, resources become popular when they are annotated by many users with popular tags, while tags, on the other hand, become popular when many users attach them to popular resources.

SocialPageRank constructs the folksonomy graph $\mathbb{G}_\mathbb{F}$ similarly to FolkRank. However, $\mathbb{G}_\mathbb{F}$ is modeled within three different adjacency matrices. A_{TR} models the edges between tags and resources. The weight

$w(t,r)$ is computed as done in the FolkRank algorithm (cf. Section 5.1.1): $w(t,r) = |\{u \in U : (u,t,r) \in Y\}|$. The matrices A_{RU} and A_{UT} describe the edges between resources and users, and users and tags respectively. $w(r,u)$ and $w(u,t)$ are again determined correspondingly. The SocialPageRank algorithm results in a vector $\vec{r}$, whose items indicate the *social PageRank* of a resource.

Definition 5 (SocialPageRank). The *SocialPageRank algorithm* (see (Bao et al., 2007)) computes a ranking of resources in folksonomies by executing the following steps:

1. **Input:** Association matrices A_{TR}, A_{RU}, A_{UT}, and a randomly chosen SocialPageRank vector $\vec{r}_0$.
2. **until** $\vec{r}_i$ converges **do**:
   (a) $\vec{u}_i = A_{RU}^T \cdot \vec{r}_i$
   (b) $\vec{t}_i = A_{UT}^T \cdot \vec{u}_i$
   (c) $\vec{r'}_i = A_{TR}^T \cdot \vec{t}_i$
   (d) $\vec{t'}_i = A_{TR} \cdot \vec{r'}_i$
   (e) $\vec{u'}_i = A_{UT} \cdot \vec{t'}_i$
   (f) $\vec{r}_{i+1} = A_{RU} \cdot \vec{u'}_i$
3. **Output:** SocialPageRank vector $\vec{r}$.

SocialPageRank and FolkRank both base on the PageRank algorithm. Regarding the underlying *random surfer model* of PageRank (Page et al., 1998), a remarkable difference between the algorithms relies on the types of links that can be followed by the "random surfer". SocialPageRank restricts the "random surfer" to paths in the form of resource-user-tag-resource-tag-user, whereas FolkRank is more flexible and allows e.g. also paths like resource-tag-resource.

5.2.2 Personalized SocialPageRank

SocialPageRank computes a global ranking of resources in folksonomies. With the *Personalized SocialPageRank* algorithm we extend SocialPageRank introduced in (Bao et al., 2007) and transform into a topic-sensitive ranking algorithm. Therefor, we introduce the ability of emphasizing weights within the input matrices of SocialPageRank so that preferences can be considered, which are possibly adapted to a certain context. For example, $w(t,r)$ is adapted as follows: $w(t,r) = pref(t) \cdot pref(r) \cdot |\{u \in U : (u,t,r) \in Y\}|$, where $pref(\cdot)$ returns the preference score of t and r respectively. The preference function $pref(\cdot)$ is specified in equation 2:

$$pref(x) = \begin{cases} 1, \text{if there is no preference in } x \\ c > 1, \text{if there is a preference in } x \end{cases} \quad (2)$$

In our evaluations (see Section 6) we utilized the Personalized SocialPageRank in order to align the SocialPageRank to the context of a keyword query t_q and specified a preference into t_q using $c = 20$.

5.2.3 GroupMe! Ranking Algorithm (GRank)

The most important application of ranking algorithms is search. In Definition 6 we introduce *GRank*, a search and ranking algorithm optimized for GroupMe! folksonomies.

Definition 6 (GRank). The *GRank* algorithm computes a ranking for all resources, which are related to a tag t_q with respect to the group structure of GroupMe! folksonomies (see Definition 3). It executes the following steps:

1. **Input:** keyword query tag t_q.
2. $\breve{R}_q = \breve{R}_a \cup \breve{R}_b \cup \breve{R}_c \cup \breve{R}_d$, where:
   (a) $\breve{R}_a$ contains resources $r \in \breve{R}$ with $w(t_q, r) > 0$
   (b) $\breve{R}_b$ contains resources $r \in \breve{R}$, which are contained in a group $g \in G$ with $w(t_q, g) > 0$
   (c) $\breve{R}_c$ contains resources $r \in \breve{R}$ that are contained in a group $g \in G$, which contains at least one resource $r' \in \breve{R}$ with $w(t_q, r') > 0$ and $r \neq r'$
   (d) $\breve{R}_d$ contains groups $g \in G$, which contain resources $r' \in \breve{R}$ with $w(t_q, r') > 0$
3. $\vec{w}_{\breve{R}_q}$ is the ranking vector of size $|\breve{R}_q|$, where $\vec{w}_{\breve{R}_q}(r)$ returns the *GRank* of resource $r \in \breve{R}_q$
4. **for each** $r \in \breve{R}_q$ **do:**
   (a) $\vec{w}_{\breve{R}_q}(r) = w(t_q, r) \cdot d_a$
   (b) **for each** group $g \in G \cap \breve{R}_a$ **do**:
   $\vec{w}_{\breve{R}_q}(r) += w(t_q, g) \cdot d_b$
   (c) **for each** $r' \in \breve{R}_a$ where r' is contained in a same
   group as r and $r \neq r'$ **do**:
   $\vec{w}_{\breve{R}_q}(r) += w(t_q, r') \cdot d_c$
   (d) **if**$(r \in G)$ **then:**
   for each $r' \in \breve{R}_a$ where r' is contained in r **do**:
   $\vec{w}_{\breve{R}_q}(r) += w(t_q, r') \cdot d_d$
5. **Output:** GRank vector $\vec{w}_{\breve{R}_q}$

$w(t_q, r)$ is the weighting function defined in Section 5.1.1 and counts the number of users, who have annotated resource $r \in \breve{R}$ with tag $t_q$in any group. When dealing with multi-keyword queries, GRank accumulates the different GRank vectors. The factors d_a, d_b, d_c, and d_d allow to emphasize the weights gained by (a) directly assigned tags, (b) tags assigned to a group the resource is contained in, (c) tags assigned to neighboring resources, and (d) tags assigned to resources of a group. For the evaluations in Section 6 we set $d_a = 10$, $d_b = 4$, $d_c = 2$, and $d_d = 4$.

Table 1: Feature overview of the different ranking strategies presented in Section 5.1 and Section 5.2.

Ranking Strategy	applicable for	topic-sensitive	group-sensitive
FolkRank (Hotho et al., 2006b)	u, t, r	yes	no
GFolkRank	u, t, r	yes	yes
GFolkRank$^+$	u, t, r	yes	yes
SocialPageRank (Bao et al., 2007)	r	no	no
Pers. SocialPageRank	r	yes	no
GRank	r	yes	yes

5.3 Synopsis

Table 1 summarizes some features of the ranking strategies presented in the previous sections. The FolkRank-based algorithms are applicable for ranking of arbitrary folksonomy entities, i.e. users, tags, and resources. Furthermore, they are topic-sensitive, which claims that they do not compute a static ranking but allow to adapt rankings to a certain context. SocialPageRank computes static, global rankings independent of the context, which is e.g. given by a keyword query. With Personalized SocialPageRank we transformed SocialPageRank into a topic-sensitive ranking algorithm. GFolkRank, GFolkRank$^+$, and GRank denote search and ranking strategies, which exploit group structures of GroupMe! folksonomies (cf. Definition 3) and are therewith *group-sensitive*.

6 EVALUATIONS

The most important application for ranking algorithms is search. Therefor, we evaluated the algorithms presented in Section 5 with respect to *search for resources* within the GroupMe! dataset, which is characterized in Section 4.1.

Topic-sensitive ranking strategies can directly be applied to the task of searching for resources, e.g. FolkRank-based algorithms can model the search query within the preference vector (see Equation 1 in Section 5.1.1) in order to compute a ranked search result list. In (Abel et al., 2008b) we evidence that our group-sensitive ranking algorithms like GFolkRank (see Section 5.1.2) improve the search and ranking quality significantly (one-tailed t-test, significance level $\alpha = 0.05$) compared to FolkRank. Non-topic-sensitive ranking strategies – like SocialPageRank – compute global, static rankings and therewith need a baseline search algorithm, which delivers a base set of possibly relevant resources, which serve as input for the ranking algorithm. In our search experiments we formulate the task to be performed by the ranking strategies as follows.

Search Task. *Given a base set of possibly relevant resources, the task of the ranking algorithm is to put these resources into an order so that the most relevant resources appear at the very top of the ranking.*

6.1 Metrics and Test Set

For evaluating the quality of the ranking strategies with respect to the *search task* we utilized the *OSim* and *KSim* metrics as proposed in (Haveliwala, 2003). $OSim(\tau_1, \tau_2)$ enables us to determine the overlap between the top k resources of two rankings, τ_1 and τ_2.

$$OSim(\tau_1, \tau_2) = \frac{|R_1 \cap R_2|}{k}, \tag{3}$$

where $R_1, R_2 \subset \check{R}$ are the sets of resources that are contained in the top k of ranking τ_1 and τ_2 respectively, and $|R_1| = |R_2| = k$.

$KSim(\tau_1, \tau_2)$ indicates the degree of pairwise distinct resources, r_u and r_v, within the top k that have the same relative order in both rankings.

$$KSim(\tau_1, \tau_2) = \frac{|\{(r_u, r_v) : \tau_1, \tau_2 \text{ agree on order of } (r_u, r_v), r_u \neq r_v\}|}{|R_{\tau_1 \cup \tau_2}| * (|R_{\tau_1 \cup \tau_2}| - 1)} \tag{4}$$

$R_{\tau_1 \cup \tau_2}$ is the union of resources of both top k rankings. When detecting whether both rankings agree on the order of two resources, we use τ_1' and τ_2'. τ_1' corresponds to ranking τ_1 extended with resources R_1' that are contained in the top k of τ_2 and not contained in τ_1. We do not make any statements about the order of resources $r \in R_1'$ within ranking τ_1'. τ_2' is constructed correspondingly.

In our analysis we apply OSim and KSim in order to compare rankings computed by the ranking strategies with optimal rankings. The optimal rankings are based on 50 hand-selected rankings: Given 10 keywords, which were out of the set of tags T, 5 experts independently created rankings for each of the keywords, which represented from their perspective the most precise top 20 ranking of resources. Therefore, they were enabled to inspect and the entire GroupMe! dataset. By building the average ranking for each keyword, we gained 10 optimal rankings. Among the 10 keywords, there are frequently used tags as well as seldom used ones.

6.2 Base Set Detection

The base set contains all search results, which are finally returned as a search result list, where the order is computed by the ranking algorithm. Hence, it is important to have a search method, which produces a base set containing a high number of relevant resources (high recall) without loosing precision. Table 2 compares different base set detection methods with each other.

Table 2: Comparison of different procedures to determine the basic set of relevant resources. Values are measured with respect to the test set described in Section 6.1.

Base Set Algorithm	*Recall*	*Precision*	*F-measure*
Basic	0.2767	0.9659	0.4301
BasicG	0.5165	0.7815	0.6220
BasicG$^+$	0.8853	0.6120	0.7237

Basic Returns only those resources, which are directly annotated with the search keyword (cf. $\breve{R}_a$ in Definition 6).

BasicG Returns in addition to *Basic* also resources, that are contained in groups annotated with the query keyword (cf. $\breve{R}_a \cup \breve{R}_b$ in Definition 6).

BasicG$^+$ This approach exploits group structures more extensively. It corresponds to our GRank algorithm without ranking the resources (cf. $\breve{R}_q$ in Definition 6).

Having a recall of nearly 90%, *BasicG$^+$* clearly outperforms the other approaches. Though the precision is lower compared to *Basic*, which searches for directly annotated resources, the *F-measure* – the weighted mean of precision and recall – certifies the decisive superiority of *BasicG$^+$*. In our experiments we thus utilize the group-sensitive *BasicG$^+$* in order to discover the set of relevant resources to be ranked. All ranking algorithms therewith benefit from the power of *BasicG$^+$*.

6.3 Experiment

In our experiment we proceed as follows. For each keyword query of our test set described in Section 6.1 and each ranking strategy presented in Section 5.1 and 5.2 we perform three steps.

1. Identification of the base set of possibly relevant resources by applying *BasicG$^+$* (see Section 6.2).
2. Execution of ranking algorithm to rank resources contained in the base set according to their relevance to the query.
3. Comparison of computed ranking with the optimal ranking of the test set by measuring *OSim* and *KSim* (see Section 6.1).

Finally, we average the OSim/Ksim values for each ranking strategy.

6.4 Results

Figures 4 and 5 present the results we obtained by running the experiments as described in the previous section. On average, the base set contains 58.9 resources and the average recall is 0.88 (cf. Table 2). The absolute OSim/KSim values are therewith influenced by the base set detection. For example, regarding the Top 20 results in Table 5, the best possible OSim value achievable by the ranking strategies is 0.92, whereas the worst possible value is 0.27, which is caused by the size and high precision of the base set. OSim and KSim both do not make any assertions about the relevance of the resources contained in the Top k. They measure the overlap of the top k rankings and the relative order of the ranked resources, respectively (see Section 6.1).

As expected, the strategy, which ranks resources randomly performs worse. However, due to the high

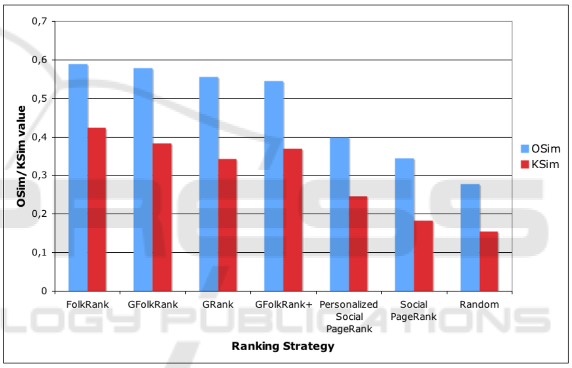

Figure 4: Top 10 OSim/KSim comparison between different ranking strategies. Basic Set is determined via BasicG$^+$ (cf. Section 6.2). Best possible OSim: 0.95. Worst possible OSim: 0.04.

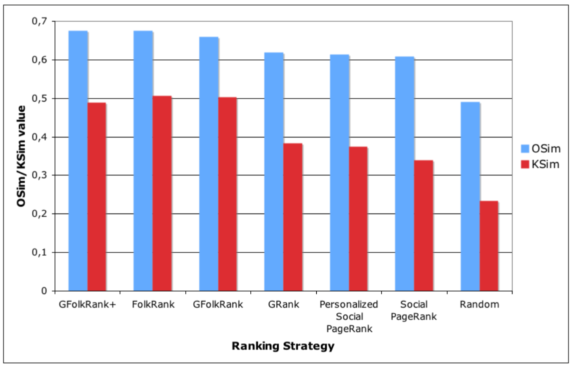

Figure 5: Top 20 OSim/KSim comparison between different ranking strategies. Basic Set is determined via BasicG$^+$ (cf. Section 6.2). Best possible OSim: 0.92. Worst possible OSim: 0.27.

quality of the group-sensitive base set detection algorithm, the performance of the random strategy is still acceptable. SocialPageRank is outperformed by the topic-sensitive ranking algorithms. Personalized SocialPageRank, the topic-sensitive version, which we developed in Section 5.2.2, improves the OSim-performance of SocialPageRank by 16% and the KSim-performance by 35%, regarding the top 10 evaluations.

The FolkRank-based strategies perform best, especially when analyzing the measured KSim values. Regarding the performance of SocialPageRank within the scope of the top 10 analysis, FolkRank, GFolkRank, and GFolkRank$^+$ improve KSim by 132%, 110%, and 102% respectively. Here, the results evaluated by the OSim metrics also indicate an increase of the ranking quality, ranging from 58% to 71%. The GRank algorithm can compete with the FolkRank-based algorithms and produces – with respect to OSim and KSim – high quality rankings as well. For example in our top 10 evaluations, GRank performs 65%/89% (OSim/KSim) better than SocialPageRank, whereas FolkRank improves GRank slightly by 5%/25% (OSim/KSim). The promising results of GRank are pleasing particularly because GRank does not require computationally intensive and time-consuming matrix operations as required by the other ranking algorithms.

The group-sensitive ranking strategies do not improve the ranking quality significantly. However, all ranking algorithms listed in Figures 4 and 5 benefit from the group-sensitive search algorithm, which determines the basic set and which supplies the best (regarding F-measure) set of resources that are relevant to the given query.

7 CONCLUSIONS

Folksonomy systems are valuable sources for improving search for Web resources. In this paper, we have described, proposed, and extended different graph-based ranking strategies for folksonomy systems, and evaluated and compared their performances with respect to ranking of search results. In addition, we analyzed the effect of using additional information about the context, in which some tagging activity took place, namely the group context provided by social systems like GroupMe!, on search and ranking. Our evaluations show that by exploiting group context we improve search performance in terms of both, recall as well as overall quality (measured via F-measure). The discussed graph-based ranking strategies overall perform very well in ranking search results. They have in common that they all adapt in one way or the other the PageRank (Page et al., 1998) ideas. However, those strategies which utilize the full folksonomy information and are topic-sensitive perform best.

REFERENCES

Abel, F., Frank, M., Henze, N., Krause, D., Plappert, D., and Siehndel, P. (2007). GroupMe! – Where Semantic Web meets Web 2.0. In *Int. Semantic Web Conference (ISWC 2007)*.

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

Abel, F., Henze, N., Krause, D., and Kriesell, M. (2008b). On the effect of group structures on ranking strategies in folksonomies. In *Workshop on Social Web Search and Mining at 17th Int. World Wide Web Conference (WWW '08)*.

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)*, pages 501–510. ACM Press.

Haveliwala, T. H. (2003). Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):784–796.

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*, pages 87–102, Aalborg.

Hotho, A., Jäschke, R., Schmitz, C., and Stumme, G. (2006b). FolkRank: A Ranking Algorithm for Folksonomies. In *Proc. of Workshop on Information Retrieval (FGIR)*, Germany.

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)*, pages 675–684. ACM Press.

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

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

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

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)*, pages 327–336. ACM Press.