Mojisola Anjorin, Thomas Rodenhausen, Renato Domínguez García, Christoph Rensing: Exploiting Semantic Information for Graph-based Recommendations of Learning Resources. In: Andrew Ravenscroft, Stefanie Lindstaedt, Carlos Kloos and Davinia Hernández-Leo: 21st Century Learning for 21st Century Skills. Proceedings of the 7th European Conference on Technology Enhanced Learning, EC-TEL 2012, vol. 7563, p. 9-22, Springer Berlin / Heidelberg, September 2012. ISBN 978-3-642-33262-3.

Exploiting Semantic Information for Graph-based Recommendations of Learning Resources

Mojisola Anjorin, Thomas Rodenhausen, Renato Domínguez García, and Christoph Rensing

Multimedia Communications Lab,
Technische Universität Darmstadt, Germany
{mojisola.anjorin,thomas.rodenhausen,renato.dominguez.garcia,christoph.rensing}@kom.tu-darmstadt.de
http://www.kom.tu-darmstadt.de

Abstract. Recommender systems in e-learning have different goals as compared to those in other domains. This brings about new requirements such as the need for techniques that recommend learning resources beyond their similarity. It is therefore an ongoing challenge to develop recommender systems considering the particularities of e-learning scenarios like CROKODIL. CROKODIL is a platform supporting the collaborative acquisition and management of learning resources. It supports collaborative semantic tagging thereby forming a folksonomy. Research shows that additional semantic information in extended folksonomies can be used to enhance graph-based recommendations. In this paper, CROKODIL's folksonomy is analysed, focusing on its hierarchical activity structure. Activities help learners structure their tasks and learning goals. AScore and AInheritScore are proposed approaches for recommending learning resources by exploiting the additional semantic information gained from activity structures. Results show that this additional semantic information is beneficial for recommending learning resources in an application scenario like CROKODIL.

Keywords: ranking, resource recommendation, folksonomy, tagging

1 Introduction

Resources found on the Web ranging from multimedia websites to collaborative web resources, become increasingly important for today's learning. Learners appreciate a learning process in which a variety of resources are used [9]. This shows a shift away from instructional-based learning to resource-based learning [17]. Resource-based learning is mostly self-directed [3] and the learner is often confronted, in addition to the actual learning process, with an overhead of finding relevant high quality learning resources amidst the huge amount of information available on the Web. In learning scenarios, recommender systems support learners by suggesting relevant learning resources [15]. An effective ranking of learning resources would reduce the overhead when learning with resources found on the Web.

Social bookmarking applications, in which users collaboratively attach tags to resources, offer support to the user during the search, annotation and sharing tasks involved in resource-based learning [3]. Tagging helps to quickly retrieve a resource later via search, or navigation, or to give an overview about the resource's content. Through the collaborative tagging of resources, a structure called a folksonomy is created. Promising results using additional semantic information to improve the ranking of resources in extended folksonomies have been made [1]. It is therefore of great interest to investigate how semantic information can benefit the ranking of learning resources in an e-learning scenario such as CROKODIL [4]. CROKODIL[1] is a platform supporting the collaborative acquisition and management of learning resources. It offers support to the learner in all tasks of resource-based learning [3]. CROKODIL is based on a pedagogical concept which focuses on *activities* as the main concept for organizing learning resources [3]. *Activities* aim to support the learner during his learning process by organizing his tasks in a hierarchical activity structure. Relevant knowledge resources found on the Web are then attached to these activities. The resulting challenge is now how best to exploit these activity structures in order to recommend relevant learning resources to other users working on related activities.

In this work, we consider the hierarchical activity structures available in the CROKODIL application scenario [4] as additional semantic information which can be used for ranking resources. We therefore propose the algorithms AScore and AInheritScore which exploit the activity structures in CROKODIL to improve the ranking of resources in an extended folksonomy for the purpose of recommending relevant learning resources.

The extended folksonomy of the CROKODIL application scenario is defined in Sect. 2. Related work is summarized in Sect. 3. Proposed approaches are implemented in Sect. 4 and evaluated in Sect. 5. This paper concludes with a brief summary and an outlook on possible future work.

2 Analysis of Application Scenario: CROKODIL

CROKODIL supports the collaborative semantic tagging [5] of learning resources thereby forming a folksonomy structure consisting of users, resources and tags [3]. Tags can be assigned tag types such as topic, location, person, event or genre. Activities as mentioned in Sect.1 are created describing learning goals or tasks to be accomplished by a learner or group of learners. Resources needed to achieve these goals are attached to these activities. In addition, CROKODIL offers social network functionality to support the learning community [3]. Groups of learners working on a common activity can be created, as well as friendship relations between two learners. In the following a folksonomy and CROKODIL's extended folksonomy are defined.

[1] `http://www.crokodil.de/`, `http://demo.crokodil.de`(retrieved 06.07.2012)

A **folksonomy** is described as a system of classification derived from collaboratively creating and managing tags to annotate and categorize content[16]. This is also known as a social tagging system or a collaborative tagging system. A folksonomy can also be represented as a folksonomy graph G_F as defined in Sect. 4.

Definition 1 (Folksonomy). *A folksonomy is defined as a quadruple [11]:* $F := (U, T, R, Y)$ *where:*

- U *is a finite set of* ***users***
- T *is a finite set of* ***tags***
- R *is a finite set of* ***resources***
- $Y \subseteq U \times T \times R$ *is a* ***tag assignment*** *relation over these sets*

E.g., user *thomas* $\in U$ attaches a tag *London* $\in T$ to the resource *olympic.org* $\in R$, thus forming a tag assignment (*thomas*, *London*, *olympic.org*) $\in Y$.

An **extended folksonomy** is a folksonomy enhanced with additional semantic information [1]. CROKODIL is an extended folksonomy where the semantic information gained from activities, semantic tag types, learner groups and friendships extend the folksonomy. These additional semantic information can also be seen as giving a context to elements in the folksonomy [4] [1]. For example, resources belonging to the same activity, can be seen as belonging to the same context of this activity.

Definition 2 (CROKODIL's Extended Folksonomy). *CROKODIL's extended folksonomy is defined as:* $F_C := (U, T_{typed}, R, Y_T, (A, <), Y_A, Y_U, G, friends)$ *where:*

- U *is a finite set of* ***learners***
- T_{typed} *is a finite set of* ***typed tags*** *consisting of pairs (t, type), where t is an arbitrary tag and type* $\in$ *{topic, location, event, genre, person, other}*
- R *is a finite set of* ***learning resources***
- $Y_T \subseteq U \times T_{typed} \times R$ *is a* ***tag assignment relation*** *over the set of users, typed tags and resources*
- $(A, <)$ *is a finite set of* ***activities*** *with a partial order* $<$ *indicating sub-activities*
- $Y_A \subseteq U \times A \times R$ *is an* ***activity assignment*** *relation over the set of users, activities and resources*
- $Y_U \subseteq U \times A$ *is an* ***activity membership assignment*** *relation over the set of users and activities*
- $G \subseteq \mathcal{P}(U)$ *is the finite set of subsets of learners called* ***groups of learners***
- $friends \subseteq U \times U$ *is a symmetric binary relation which indicates a* ***friendship relation*** *between two learners*

E.g., *thomas* is preparing for a quiz about the olympic games. He therefore creates an activity *prepare quiz about the olympics* having a sub-activity *collect historical facts*. This means A = {*prepare quiz about the olympics*, *collect historical facts*} and *collect historical facts* $<$ *prepare quiz about the olympics*.

In addition, (*thomas*, *prepare quiz about the olympics*) $\in Y_U$ and (*thomas*, *collect historical facts*) $\in Y_U$. He finds the website *olympic.org*, to which he attaches the tag *London* with tag type *location*, (*thomas*, (*London*, *location*), *olympic.org*) $\in Y_T$. He then attaches this resource to the activity *prepare quiz about the olympics*, (*thomas*, *prepare quiz about the olympics*, *olympic.org*) $\in Y_A$. Thomas creates a group *olympic experts* $\in G$ and invites *moji* $\in U$ and his friend *renato* $\in U$ to help him gather facts about the olympic games.

In this paper, we will be focusing on the additional semantic information gained from the activities in CROKODIL's extended folksonomy and investigating how this can improve the ranking of learning resources.

3 Related Work

Recommender systems have shown to be very useful in e-learning scenarios [15]. Collaborative filtering approaches use community data such as feedback, tags or ratings from learners to make recommendations e.g.[8] whereas content-based approaches make recommendations based on the similarity between learning resources e.g. [18]. Recommender systems in e-learning have different information retrieval goals as compared to other domains thus leading to new requirements like recommending items beyond their similarity [15]. It is therefore increasingly important to develop recommender systems that consider the particularities of the e-learning domain. Graph-based recommendation techniques can be classified as neighborhood-based collaborative filtering approaches, having the advantage of avoiding the problems of sparsity and limited coverage [7]. Graph-based recommender systems e.g. [1, 6] consider the graphical structure when recommending items in a folksonomy. The data is represented in the form of a graph where nodes are users, tags or resources and edges the transactions or relations between them. One of the most popular approaches is FolkRank [12] which is based on the PageRank computation on a graph created from a folksonomy. FolkRank can be used to recommend users, tags or resources in social bookmarking systems. The intuition is that a resource tagged with important tags by important users becomes important itself. The same holds for tags and users.

Furthermore, it is of interest for recommender systems in e-learning to take advantage of additional semantic information such as context awareness which includes pedagogical aspects like learning goals [15]. Abel [1] shows it is worth exploiting additional semantic information which are found in extended folksonomies to improve ranking strategies. Approaches, for example GFolkRank [1], are introduced which extend FolkRank to a context-sensitive ranking algorithm exploiting the additional semantic information gained from the grouping of resources in GroupMe![2]. Groups in GroupMe! allow resources e.g. belonging to a common topic to be semantically grouped together. Groups can also contain other groups [2]. GFolkRank, an extension of FolkRank [12] is a ranking algorithm that leverages groups available in GroupMe! for ranking. Groups are

[2] `http://groupme.org/`, retrieved 06/07/2012

interpreted as tags i.e. if a user adds a resource r to a group g then GFolkRank translates this as a tag (group) assignment. The folksonomy graph is therefore extended with additional group nodes and group assignments. In addition, other approaches are proposed such as GRank [1]. GRank is designed for ranking resources with a tag as input. It computes a ranking for all resources, which are related to the input tag with respect to the group structure in GroupMe!

The concept of groups in the GroupMe! application is similar to the concept of activities in the CROKODIL application. Therefore, this opportunity to exploit the semantic information gained from activities in CROKODIL will be investigated in the following sections.

4 Concept and Implementation

Given a certain user u as input, the resource recommendation task is to find a resource r which is relevant to this user. This recommendation task is also seen as a ranking task. A ranking algorithm computes for an input user u a score vector that contains the score values $\boldsymbol{score}(r)$ for each resource r in the graph. These scored resources are then ordered forming a ranked list according to their score values with the highest scored resource at the top of the list. The top ranked resources are then recommended to the user u. For example, the scores $\boldsymbol{score}(r_1) = 5$ and $\boldsymbol{score}(r_2) = 7$ and $\boldsymbol{score}(r_3) = 3$ create a ranked list: r_2, r_1 and r_3. Therefore the top recommendation to user u will be resource r_2.

We propose two ranking algorithms, AScore and AInheritscore. Both algorithms compute a folksonomy graph G_F considering not only activities when ranking resources but also including activity hierarchies and users assigned to work on these activities in the graph structure.

In the following, three sets are defined that will be used in Definition 3 to determine the weights of the edges in the folksonomy graph G_F. For a given user $u \in U$, tag $t \in T$ and resource $r \in R$:

– Let $U_{t,r} = \{\ u \in U \mid (u,t,r) \in Y\ \} \subseteq U$ be the set of all users that have assigned resource r a tag t
– Let $T_{u,r} = \{\ t \in T \mid (u,t,r) \in Y\ \} \subseteq T$ be the set of all tags that user u assigned to resource r
– Let $R_{u,t} = \{\ r \in R \mid (u,t,r) \in Y\ \} \subseteq R$ be the set of all resources that user u assigned a tag t

Definition 3 (Folksonomy Graph). *Given a folksonomy F, the folksonomy graph G_F [1] is defined as an undirected, weighted graph $G_F := (V_F, E_F)$ where:*

– $V_F = U \cup T \cup R$ *is the set of nodes*
– $E_F = \{\ \{u,t\}\ ,\ \{t,r\}\ ,\ \{u,r\} \mid u \in U, t \in T, r \in R, (u,t,r) \in Y\ \} \subseteq V_F \times V_F$ *is the set of undirected edges*
– *Each of these edges is given a weight $w(e), e \in E_F$ according to their frequency within the set of tag assignments:*
  • $w(u,t) = |R_{u,t}|$ *the number of resources that user u assigned the tag t*
  • $w(t,r) = |U_{t,r}|$ *the number of users who assigned tag t to resource r*
  • $w(u,r) = |T_{u,r}|$ *the number of tags that user u assigned to resource r*

4.1 AScore

AScore is an algorithm based on GFolkRank [1] as described in Sect. 3. AScore extends the folksonomy graph G_F in a similar way with activity nodes and activity assignments. However, in addition, AScore extends the folksonomy graph with activity hierarchy relations between activities (4) as well as with users belonging to an activity (3). A user u is said to belong to an activity a, when the user u is working on the activity a. This is represented as an edge in the graph between u and a. Furthermore, AScore considers the hierarchical activity structure when determining the weights of the newly introduced edges. The AScore algorithm is described below:

- Let $G_C = (V_C, E_C)$ be the folksonomy graph of the extended folksonomy F_C
- $V_C = V_F \cup A$
- E_C is a combination of edges (1) from the folksonomy graph E_F with E_A (2), which are all activity assignments where a user u added a resource r to an activity a. Additionally, E_U (3) is added, which comprises all assignments of a user u to an activity a. Finally, the activity hierarchies E_H (4) are added as edges between a sub-activity a_{sub} and a super-activity a_{super}.

$$E_C = E_F \cup \ E_A \cup \ E_U \cup \ E_H \ . \tag{1}$$

$$E_A = \{\{u,a\},\{a,r\},\{u,r\} \mid u \in U, r \in R, a \in A, (u,a,r) \in Y_A\} \ . \tag{2}$$

$$E_U = \{\{u,a\} \mid u \in U, a \in A, (u,a) \in Y_U\} \ . \tag{3}$$

$$E_H = \{\{a_{sub}, a_{super}\} \mid a_{sub}, a_{super} \in A, a_{sub} < a_{super}\} \ . \tag{4}$$

The newly introduced edges are now given weights. The edges in E_A are given all the same weight $activityAssign(u,r,a)$ (5) because, similar to GFolkRank [1], a resource can only be added once to an activity. Attaching additional semantic information to a resource (like assigning it to a group in GroupMe! or to an activity in CROKODIL) is seen as more valuable than simply tagging it [1], therefore $activityAssign(u,r,a)$ is assigned the maximum number of users who assigned tag t to resource r (5). Similarly, the edges between a user u and an activity a are given the weight $w_{Membership}(u,a)$ (6) which is the maximum number of resources assigned with tag t by user u, who is working on activity a. The edges between activities of the same hierarchy are given the weight $w_{Hierarchy}(a_{sub}, a_{super})$. These edges are seen to be at least as strong as the connections between an activity and other nodes in the graph, therefore in (7), the maximum weight is assigned.

$$\begin{aligned} w(u,a) = w(a,r) = w(u,r) = activityAssign(u,r,a) \\ \text{where } activityAssign(u,r,a) = max(\mid U_{t,r} \mid) \ . \end{aligned} \tag{5}$$

$$w_{Membership}(u,a) = max(\mid R_{u,t} \mid) \ . \tag{6}$$

$$w_{Hierarchy}(a_{sub}, a_{super}) = max(activityAssign(u,r,a_{sub}), w_{Membership}(u,a_{sub})) \ . \tag{7}$$

After the folksonomy graph G_C has been created and the weights of the edges determined, any graph-based ranking algorithm for folksonomies e.g. FolkRank can now be applied to calculate the scores of each node.

4.2 AInheritScore

AInheritScore is an algorithm based on GRank [1] as described in Sect. 3. AInheritscore computes for an input user u a score vector that contains the score values $\boldsymbol{score}(r)$ for each resource r. The input user u however needs to be transformed into input tags t_q, depending upon how many tags the user u has. Each of these input tags t_q is weighted according to its frequency of usage by user u.

The parameters d_a, d_b, d_c are defined to emphasize the "inherited" scores gained by relations in the hierarchy. The values of these parameters are set in Sect. 5 for the evaluations.

1. d_a for resources having the input tag directly assigned to them
2. d_b for resources in the activity hierarchy having a resource that is tagged with the input tag
3. d_c for users in the activity hierarchy having assigned the input tag

Additionally, an activity distance $activityDist(a_1, a_2)$ between two activities is calculated as the number of hops from activity a_1 to activity a_2. However, it is also possible to calculate a lesser distance for sub-activities, or include the fan-out in the computation. AInheritscore contrasts to GRank in the following points:

1. Activities are not considered to be resources and cannot be assigned a tag.
2. AInheritscore considers activity hierarchies as well as users assigned to activities when computing the scores.
3. Activity hierarchies are leveraged by the inheritance of scores. These scores are emphasized by considering the connections in the activity hierarchy. The distance between activities in the hierarchy are considered as well.

AInheritscore algorithm is described in the following steps:

1. For each input tag t_q
2. Let $\boldsymbol{score} = \mathbf{0}$ be the score vector
3. Determine $R_q = R_a \cup R_b \cup R_c$ where:
   (a) R_a contains all resources with the input tag t_q directly assigned to them $w(t_q, r) > 0$.
   (b) R_b contains all resources belonging to the same activity hierarchy as another resource r, that has the input tag t_q directly assigned to it: $w(t_q, r) > 0$
   (c) R_c contains all resources belonging to the same activity hierarchy as a user u, who has tagged a resource with the input tag tq: $w(u, t_q) > 0$
4. For all $r \in R_q$ belonging to activity a do
   (a) increase the score value of r:

$$\boldsymbol{score}(r) += w(t_q, r) \cdot d_a \tag{8}$$

(b) for each $r' \in R_q$ belonging to activity a', where a' and a are in the same activity hierarchy, increase again the score of r:

$$\boldsymbol{score}(r) += \frac{w(t_q, r')}{activityDist(a, a')} \cdot d_b \tag{9}$$

(c) for each $u \in U_q$ working on activity a', where a' and a are in the same activity hierarchy, increase again the score of r:

$$\boldsymbol{score}(r) += \frac{w(u, t_q)}{activityDist(a, a')} \cdot d_c \tag{10}$$

5. Output: $\boldsymbol{score}$

5 Evaluation

The goal of this paper is to investigate how the implicit semantic information contained in activity hierarchies can be exploited to improve the ranking of resources in an extended folksonomy such as CROKODIL. As the CROKODIL data set has not yet attained a sufficient size for significant evaluation, a data set with an extended folksonomy containing similar concepts to those of activities in CROKODIL was sought.

5.1 Corpus

The GroupMe! data set was chosen as the concept of groups in GroupMe! is a similar concept to the activities and activity hierarchies in CROKODIL as mentioned in Sect. 3. There are however differences and a mapping of the concepts is necessary to be able to use the data set:

- The aim of groups in GroupMe! is to provide a collection of related resources. In CROKODIL however, activities are based on a pedagogical concept to help learners structure their learning goals in a hierarchical structure. Learning resources needed to achieve these goals are attached to these activities. Therefore, the assignment of a resource to a group in GroupMe! is interpreted as attaching a resource to an activity in CROKODIL.
- Groups in GroupMe! are considered resources and can therefore belong to other groups. These groups of groups or hierarchies of groups are interpreted as activity hierarchies in CROKODIL.
- Tags can be assigned to groups in GroupMe!. In contrast however, tags can not be assigned to activities in CROKODIL. These tags on groups in GroupMe! are therefore not considered in the data set.

Groups of groups or group hierarchies are unfortunately sparse in the GroupMe! data set. A p-core extraction [13] would reduce these hierarchies even more, therefore no p-core extraction is made. The data set has the characteristics described in Table 1.

Table 1. The extended folksonomy GroupMe! data set

Users	Tags	Resources	Groups	Posts	Tag Assignments
649	2580	1789	1143	1865	4366

5.2 Evaluation Methodology

The evaluation methodology **LeavePostOut** [13] is used for the evaluations of AScore and AInheritscore. In addition, we propose an evaluation methodology **LeaveRTOut** which is inspired from LeavePostOut. A post $P_{u,r}$ is defined in [11] as all tag assignments of a specific user u to a specific resource r. LeavePostOut as shown in Fig.1 removes the post $P_{u,r}$, thereby ensuring that no information in the folksonomy remains that could connect the user u directly to resource r [13]. LeaveRTOut as shown in Fig. 2 eliminates the connection in the

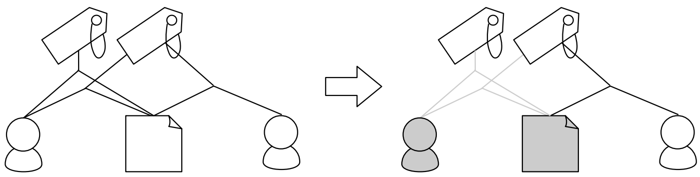

Fig. 1. LeavePostOut evaluation methodology

folksonomy between a tag t and a resource r instead of eliminating the connection between a user u and a resource r. LeaveRTOut therefore sets a different task to solve as LeavePostOut. For the evaluations, the user u of a post is used

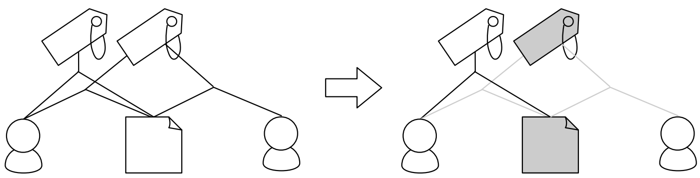

Fig. 2. LeaveRTOut evaluation methodology

as input. LeavePostOut is used to determine adequate parameters for the algorithms. AInheritScore takes the values of GRank's parameters which according to a sensitivity analysis in [1] shall be set to $d_a = 10$, $d_b = 2$. d_c is set as well as $d_b = d_c = 2$.

For the evaluations, the metrics *Mean Average Precision (MAP)* and *Precision at k* [14] are used. MAP is used to determine the overall ranking quality while Precision at k determines the ranking quality in the top k positions. Precision at k is extended to *Mean Normalized Precision (MNP) at k* to obtain a single measure over a number of information needs Q as well as to be more suitable for the evaluation methodology, i.e. in respect to the maximal achievable $Precision_{max}(k)$. Mean Normalized Precision at k is defined as follows:

$$MNP(Q,k) = \frac{1}{|Q|} \cdot \sum_{j=1}^{|Q|} \frac{Precision(k)}{Precision_{max}(k)} \qquad (11)$$

For the statistical significant tests, *Average Precision* [14] is used for a single information need q, applying the *Wilcoxon signed-rank tests* [3].

5.3 Results

LeavePostOut and LeaveRTOut results from AScore and AInheritScore are compared to those of GRank, GFolkRank, FolkRank and Popularity. Popularity is calculated as the number of tags and users a resource is connected to. The results are visualized as a violin plot [10] in Fig.3 and Fig.4. The distribution of the data

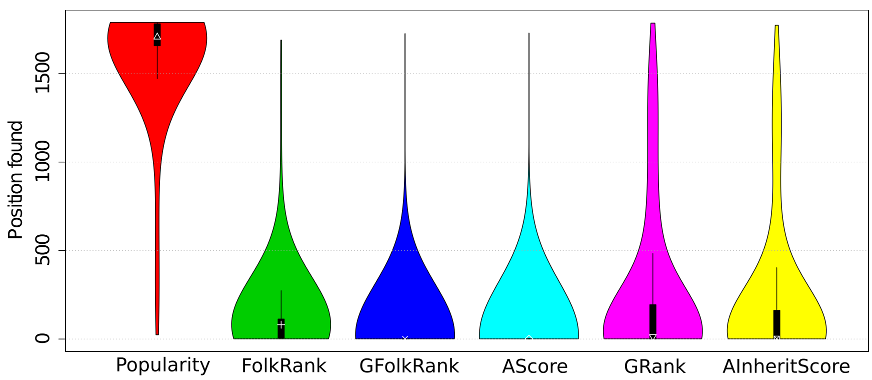

Fig. 3. Violinplot of LeavePostOut results

values are shown along the y-axis. The width of the violin plot is proportional to the estimated density at that point. As can be seen, most of the algorithms have most items ranked in positions < 500, whereas popularity still has too many items ranked in further positions. The Mean Average Precision results for LeavePostOut are shown in Table 2. GFolkRank and AScore perform best with regard to the overall ranking effectiveness. Both achieve a MAP of 0.70, followed by AInheritScore, GRank, FolkRank and last Popularity. The MAP results for LeaveRTOut are presented in Table 3. GFolkRank and AScore perform best with a MAP of 0.20, followed by FolkRank, GRank, AInheritScore and last Popularity. The results of the Mean Normalized Precision at k for $k \in [1, 10]$ for both LeavePostOut (left) and LeaveRTOut (right) are shown in Fig.5. The results

[3] `http://stat.ethz.ch/R-manual/R-patched/library/stats/html/wilcox.test.html`, retrieved 20/03/2012

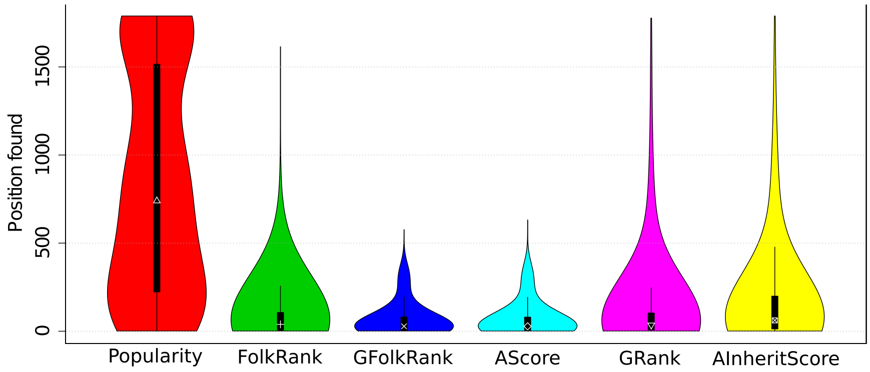

Fig. 4. Violinplot of LeaveRTOut results

Table 2. Mean Average Precision (MAP) results for LeavePostOut

Popularity	FolkRank	GFolkRank	AScore	GRank	AInheritscore
0,00	0,19	0,70	0,70	0,38	0,47

Table 3. Mean Average Precision (MAP) results for LeaveRTOut

Popularity	FolkRank	GFolkRank	AScore	GRank	AInheritscore
0.02	0.18	0.20	0.20	0.14	0.11

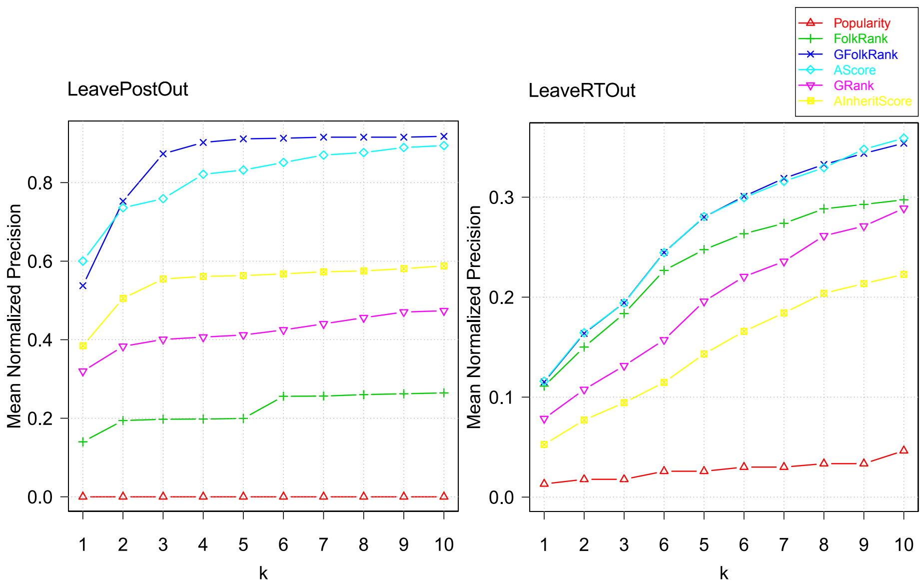

Fig. 5. Mean Normalized Precision at k: LeavePostOut (left) and LeaveRTOut(right)

of all pairwise comparisons for statistical significance are shown in Table 4 and Table 5. The LeavePostOut results differ from the LeaveRTOut results due to the fact that they set a differently hard task to solve. Hence, the results from the two methodologies are useful to assess the effectiveness of the algorithms in different ranking scenarios. For example, results from LeavePostOut show on the one hand, that GFolkRank is more effective than AScore. On the other hand, results from LeaveRTOut show that AScore is more effective than GFolkRank. In summary, LeavePostOut results show that the algorithms leveraging additional semantic information are overall more effective than FolkRank as these algorithms designed for the extended folksonomy have the advantage of being able to leverage the additional information gained from activities to recommend relevant resources. The selection of an algorithm for ranking learning resources will therefore depend upon its application scenario and what is important for ranking. For example, AScore would be the choice when activity hierarchies are particularly important for ranking learning resources such as in the CROKODIL application scenario or GFolkRank if this is not the case in other scenarios.

Table 4. Significance matrix of pair-wise comparisons of LeavePostOut results

More effective than →	Popularity	FolkRank	GFolkRank	AScore	GRank	AInheritScore
Popularity	☐	☐	☐	☐	☐	☐
FolkRank	⊠	☐	☐	☐	☐	☐
GFolkRank	⊠	⊠	☐	⊠	⊠	⊠
AScore	⊠	⊠	☐	☐	⊠	⊠
GRank	⊠	⊠	☐	☐	☐	☐
AInheritScore	⊠	⊠	☐	☐	⊠	☐

Table 5. Significance matrix of pair-wise comparisons of LeaveRTOut results

More effective than →	Popularity	FolkRank	GFolkRank	AScore	GRank	AInheritScore
Popularity	☐	☐	☐	☐	☐	☐
FolkRank	⊠	☐	☐	☐	⊠	⊠
GFolkRank	⊠	⊠	☐	☐	⊠	⊠
AScore	⊠	⊠	⊠	☐	⊠	⊠
GRank	⊠	☐	☐	☐	☐	⊠
AInheritScore	⊠	☐	☐	☐	☐	☐

Limitations The proposed algorithms AScore and AIhneritscore are fundamentally based on the concept of activity hierarchies from the CROKODIL application scenario. The results achieved with the GroupMe! data set thus may not be representative as the group hierarchies from the GroupMe! data set modeled as the CROKODIL activity hierarchies were very sparse. Furthermore, the parameters for the algorithms were based on MAP values from LeavePostOut with a user as input. The algorithms may perform differently with regard to a metric or evaluation methodology, if parameterized accordingly. Additionally, the statistical significance is computed based on Average Precision, which is a measure of the overall ranking quality. If the statistical significance is to be compared based on the effectiveness of ranking in top positions, a different series of significance tests needs to be conducted.

6 Conclusion

Resource-based learning is mostly self-directed and the learner is often confronted with an overhead of finding relevant high quality learning resources on the Web. Graph-based recommender systems that recommend resources beyond their similarity can reduce the effort of finding relevant learning resources. We therefore propose in this paper two approaches AScore and AInheritScore that exploit the hierarchical activity structures in CROKODIL to improve the ranking of resources in an extended folksonomy for the purpose of recommending learning resources. Evaluation results show that this additional semantic information is beneficial for recommending learning resources in an application scenario such as CROKODIL. The algorithms leveraging additional semantic information are overall more effective than FolkRank as these algorithms designed for the extended folksonomy have the advantage of being able to leverage the additional information gained from activities and activity hierarchies to recommend relevant resources.

Future work will be to evaluate these algorithms with a data set from the CROKODIL application scenario. Additionally, a user study in the CROKODIL application scenario is planned to determine the true relevance of recommendations of learning resources based on human judgement in a live evaluation.

Acknowledgments.

This work is supported by funds from the German Federal Ministry of Education and Research and the mark 01 PF 08015 A and from the European Social Fund of the European Union (ESF). The responsibility for the contents of this publication lies with the authors.

Special thanks to Fabian Abel for making the most current data set from GroupMe! available for the evaluations.

References

1. Abel, F.: Contextualization, User Modeling and Personalization in the Social Web. PhD Thesis, Gottfried Wilhelm Leibniz Universitt Hannover (2011)

2. Abel, F., Frank, M., Henze, N., Krause, D., Plappert, D., Siehndel, P.: GroupMe! - Where Semantic Web Meets Web 2.0. In: Proceedings of the 6th International Semantic Web Conference and 2nd Asian Semantic Web Conference on the Semantic Web. pp. 871–878 (2007)
3. Anjorin, M., Rensing, C., Bischoff, K., Bogner, C., Lehmann, L., Reger, A., Faltin, N., Steinacker, A., Lüdemann, A., Domínguez García, R.: CROKODIL - A Platform for Collaborative Resource-Based Learning. In: EC-TEL 2011. pp. 29–42 (2011)
4. Anjorin, M., Rensing, C., Steinmetz, R.: Towards ranking in folksonomies for personalized recommender systems in e-learning. In: Proc. of the 2nd Workshop on Semantic Personalized Information Management: Retrieval and Recommendation. vol. 781, pp. 22–25. CEUR-WS (Oct 2011)
5. Böhnstedt, D., Scholl, P., Rensing, C., Steinmetz, R.: Collaborative semantic tagging of web resources on the basis of individual knowledge networks. In: UMAP. LNCS (2009)
6. Cantador, I., Konstas, I., Jose, J.: Categorising Social Tags to Improve Folksonomy-Based Recommendations. Web Semantics: Science, Services and Agents on the World Wide Web 9, 1–15 (2011)
7. Desrosiers, C., G., K.: A Comprehensive Survey of Neighborhood-Based Recommendation Methods. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P. (eds.) Recommender Systems Handbook, pp. 107–144. Springer (2011)
8. Drachsler, H., Pecceu, D., Arts, T., Hutten, E., Rutledge, L., van Rosmalen, P., Hummel, H.G.K., Koper, R.: Remashed - recommendations for mash-up personal learning environments. In: Cress, U., Dimitrova, V., Specht, M. (eds.) EC-TEL 2009. LNCS, vol. 5794, pp. 788–793. Springer (2009)
9. Hannafin, M., Hill, J.: Resource-Based Learning. Handbook of Research on Educational Communications and Technology pp. 525–536 (2008)
10. Hintze, J., Nelson, R.: Violin plots: A box plot-density trace synergism. The American Statistician 52(2), 181–184 (1998)
11. Hotho, A., Jschke, R., Schmitz, C., Stumme, G.: BibSonomy: A Social Bookmark and Publication Sharing System. In: Proc. of the Conceptual Structures Tool Interoperability Workshop (2006)
12. Hotho, A., Jschke, R., Schmitz, C., Stumme, G.: Information Retrieval in Folksonomies: Search and Ranking. In: Proc. of the 3rd European Semantic Web Conference on the Semantic Web: Research and Applications. pp. 411–426 (2006)
13. Jäschke, R., Marinho, L., Hotho, A., L., S.T., G., S.: Tag Recommendations in Folksonomies. In: Proc. of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases. pp. 506–514 (2007)
14. Manning, C., Raghavan, P., Schtze, H.: Introduction to Information Retrieval. Cambridge University Press (2008)
15. Manouselis, N., Drachsler, H., Vuorikari, R., Hummel, H., Koper, R.: Recommender Systems in Technology Enhanced Learning. In: Ricci, F., Rokach, L., Shapira, B., Kantor, P. (eds.) Recommender Systems Handbook, pp. 387–415. Springer (2011)
16. Peters, I.: Folksonomies: Indexing and Retrieval in Web 2.0. De Gruyter Saur (2010)
17. Rakes, G.: Using the Internet as a Tool in a Resource-Based Learning Environment. Educational Technology 36, 52–56 (1996)
18. Romero Zaldivar, V., Crespo Garca, R., Burgos, D., Kloos, C., Pardo, A.: Automatic discovery of complementary learning resources. In: Kloos, C., Gillet, D., Crespo Garca, R., Wild, F., Wolpers, M. (eds.) EC-TEL 2011, LNCS, vol. 6964, pp. 327–340. Springer (2011)