Ranking Resources in Folksonomies by Exploiting Semantic Information

Thomas Rodenhausen
Multimedia Communications Lab
Technische Universität Darmstadt, Germany
thomas.rodenhausen@kom.tu-darmstadt.de

Mojisola Anjorin
Multimedia Communications Lab
Technische Universität Darmstadt, Germany
mojisola.anjorin@kom.tu-darmstadt.de

Renato Domínguez García*

Christoph Rensing†

Ralf Steinmetz‡

ABSTRACT

Organizing and sharing resources are the main aims of social bookmarking applications. By tagging resources, folksonomies emerge collaboratively. Information in folksonomies is valuable for ranking resources in social bookmarking applications as well as on the Web. Folksonomies are therefore important for knowledge management. There are however limitations to ranking in folksonomies, as they are not designed for search. As new Web 2.0 applications emerge providing semantic information, it becomes essential to incorporate this information for improved ranking strategies. Hence, in this work, the algorithms AspectScore and InteliScore are proposed. Both algorithms aim to overcome limitations and drawbacks of graph-based ranking algorithms in folksonomies by incorporating semantic information. Furthermore, a method that leverages semantic information to disambiguate tags is proposed as well as an evaluation methodology for ranking resources in folksonomies.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Information filtering, Retrieval models, Search process*; H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Linguistic processing*

General Terms

Algorithms, Human Factors, Experimentation, Measurement

*Author is affiliated with the same institution
†Author is affiliated with the same institution
‡Author is affiliated with the same institution

12th International Conference on Knowledge Management and Knowledge Technologies 2012 Graz, Austria

Keywords

Tagging, Folksonomy, Ranking, Recommendation

1. INTRODUCTION

Tagging, the assignment of a term by a user to a resource, plays an increasingly important role in today's knowledge management. Social bookmarking applications are often used to organize knowledge resources. The information created while tagging helps the tagger to classify and manage his resources. Depending on the tag, the classification can be quite different in nature. A resource can be tagged as *Web 2.0* or *Linguistics*, *interesting* or *boring*, *Graz* or *the outer space*. Based on tags, a user can thus, besides sharing resources with others, also give his opinion or share his knowledge about a topic [6]. The collaborative tagging of resources creates a folksonomy [19]. The information shared with others by means of tagging can also help to retrieve resources via search, navigation, or to give an overview about their content. With a high number of users in such systems, the wisdom of the crowds effect [23] allows them to deliver relevant and authoritative results. This is where ranking algorithms, e.g. used in recommender systems, can provide great benefit to the users. Ranking algorithms rank resources according to certain criteria e.g. their relevance to an information need. Hence, by means of ranking, interesting resources are recommended to users. Another benefit of ranking lies in search, where the user only has to scan a certain amount of top ranked resources to find relevant information.

However, the creation of such a ranking is not a trivial task as folksonomies are not designed for search [19]. The information about resources may be sparse e.g. due to a lack of semantic information. Hence, it is inevitable to include such information in order to create high-quality ranking. This work thus investigates resource ranking in folksonomies by exploiting semantic information.

2. RELATED WORK

The resource ranking tasks for this work are defined as follows as adapted from [5]. *Interests match* uses a user, *guided search* a tag, and *more like this* a resource as query entity. Combinations of these tasks, due to a set of query entities,

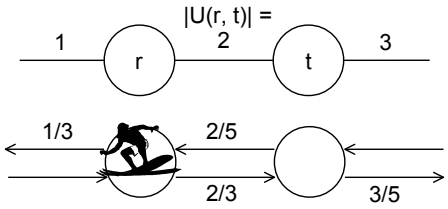

Figure 1: Probabilistic graph traversal in FolkRank

are additionally possible. In the literature, resource ranking in folksonomies has been studied for different ranking tasks using techniques that leverage different information sources. Hotho et al. propose FolkRank [13], an adaptation of PageRank [7], which can be used for resource ranking independent of the type of query entity. As the folksonomy is represented as a graph in FolkRank, it is often referred to as graph-based recommendation [20]. Graph-based recommendation in folksonomies is a form of collaborative recommendation [20] as it makes use of the individual knowledge of a user as well as the social knowledge. FolkRank's computation can be illustrated as a surfer on the folksonomy graph. The entities of the folksonomy are nodes of the graph. The surfer probabilistically traverses the graph depending on the edge weights of the undirected edges as shown in Figure 1. The edge weights between a resource and a tag are determined by the number of users that attached the tag to the resource $|U(r,t)|$. Likewise for edges between other types of entities. The more often a node is visited by the surfer, the higher it appears in the ranking. The visit rate is said to be the node's and hence the respective entity's score. The union of all nodes' scores forms the set of scored entities. This computation follows that of the random surfer model [7] of PageRank. Therefore, in order to measure relevance, FolkRank makes the Assumption 1 about the folksonomy content and structure, as extended from [1]. In response to a query, the surfer jumps with a certain probability to the nodes that represent the set of query entities, which corresponds to the biased surfer model of PageRank [10].

Assumption 1.

(i) Tags assigned to a resource describe the resource's content well.
(ii) Resources a tag is assigned to describe the tag's semantic well.
(iii) Tags assigned by a user describe the user's interests well.
(iv) Users that assigned a tag describe the tag's semantic well.
(v) A user's resources describe the user's interests well.
(vi) Users of a resource describe the resource's content well.

Peters argues that folksonomies lack a quality control of tags for a resource, e.g. by means of a controlled vocabulary [19]. Ames et al. [2] find that users have different intentions to tag e.g. for later retrieval, as notes or representing opinion. Not all tags are related to the content of resources. Therefore, tags may or may not benefit collaborative recommendation. Böhnstedt et al. however argue that users have a concept in mind while tagging [6]. Hence, even with the intention of using the tag for later retrieval, tags may be categorized into different concept types describing an aspect of a resource e.g. a tag *Barcelona* of type *Location* may describe the location in which a user got to know the resource, whereas the same tag of type *Topic* describes its content. Böhnstedt et al. suggest that a folksonomy semantically enhanced by these tag

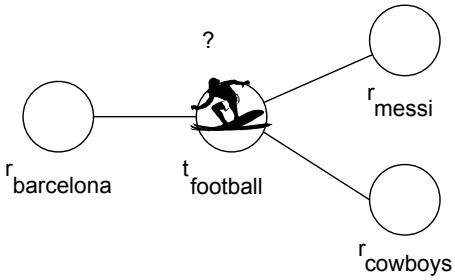

Figure 2: Concept drift due to ambiguity of *football*

types can improve recommendation in folksonomies [6]. Another kind of problem is ambiguity. Peters [19] describes the tagspace as noisy due to the homonymy problem. Therefore, she argues that folksonomies are not suited for precision in tag-based search. Au Yeung et al. [4] propose the disambiguation of tags in folksonomies by clustering the folksonomy graph. Abel et al. avoid the homonymy and synonymy problem by having the user attach the semantic to a tag [1]. In contrast to Abel et al., the disambiguation of tags in this paper is not investigated on the level of linguistic knowledge of semantics but on the level of pragmatics. For ranking, Abel et al. show that a freely given hypernym or the described attachment of a semantic concept of a tag may benefit ranking in folksonomies [1] and Cantador et al. show that some categories of tags allow for better ranking than others [8].

3. CONCEPT AND REALIZATION

As described previously, users have a concept in mind while tagging [6], which describes an aspect of a resource for a user. Hence, tag types describing this aspect provide information on the level of pragmatics of linguistic knowledge. The formalized definition of a folksonomy as adapted from [13] is extended by tag types in Definition 1:

Definition 1. A folksonomy extended by tag types is a tuple $F_A = \{U, T, A, R, Y\}$ where U, T, A and R are finite sets, whose elements are called users, tags, tag types, and resources, respectively. Y is the quaternary tag assignment relation between them $Y \in U \times T \times A \times R$.

In this work $A = \{Topic, ResourceType, Person/Organization, Location, Event, Activity, Other\}$. Another source of semantic information is the semantic relatedness of tags [22]. There exist several semantic relatedness measures e.g. XESA [22]. Tags in many folksonomy applications can often be created without restrictions. They are thus sometimes words stuck together, abbreviations, the stem of a word or even neologisms. A word, used as tag, may be modified by inflection or creativity. The usage of a tokenizer, stemmer or lemmatizer is useful, though with limitations. Hence, it remains a challenge to determine the semantic relatedness between tags.

The challenges of graph-based ranking in folksonomies are described in the context of the computation of FolkRank as the described surfer on the folksonomy graph. The surfer's choice which node to visit next, depends solely on the existing tag assignments in a folksonomy. This may introduce concept drift which harms the measurement of relevance. Figure 2 illustrates the challenge of concept drift. The surfer may intend to visit nodes related to the semantic concept of *soccer*. However, he may drift off to nodes related to the semantic concept of *american football*, as the tag *football* is

ambiguous and connected to resources related to both the semantic concepts of *soccer* and *american football*. Jurafsky et al. describe ambiguity as alternative linguistic structures for a given input [15]. For a single term as input, there may be, as in the example above, an alternative semantic of the term by means of homonymy.

However, more tasks other than word sense disambiguation (which is on the level of semantic linguistic knowledge), can be viewed as ambiguity resolving tasks [15]. The synonymy problem is about words with the same meaning but different spelling. Ambiguity may be on the linguistic level of pragmatics [15] e.g. a tag *Barcelona* may be used to describe the content of a resource, as well as other aspects relevant to a user, e.g. the *Activity* of traveling to Barcelona. Tag types can thus help to alleviate concept drift caused by the ambiguity of tags. In the folksonomy extended by tag types (Definition 1), tags are disambiguated with respect to different aspects of a resource that users may describe by tagging. This disambiguation, however, is limited to and performed on the pragmatics level of linguistic knowledge. For example, there may be a tag *football* of tag type *Topic* and of tag type *Activity*. The described scenario can be disambiguated using tag types. However, a tag *football* representing both semantic concepts *soccer* and *american football* and having for each the tag type *Topic* can not be disambiguated. Moreover, semantic relatedness can help to alleviate concept drift caused by the ambiguity of tags. Assume two synonym tags are connected to the same resource and imagine a surfer comes from one of these tags to the resource. The semantic relatedness measure is ideally the maximal value between these two tags. Hence, it is possible to reduce concept drift by attenuating the connections to other tags connected to the resource.

Furthermore, the violation of Assumption 1, e.g. tags may not describe a resource well, may introduce concept drift as users may use tags to express e.g. opinions [6]. Tag types can allievate this problem by having the surfer focus on connections related to tags of type *Topic*, as they describe the content of resources. On the other hand, however, opinionated or sentimental tags may be leveraged to assess the quality of a resource independent of a query. Semantic relatedness can alleviate this problem by having the surfer focus on connections related to tags which are semantically stronger related.

The multi-facetedness of entities in the folksonomy is another reason that may introduce concept drift, e.g. a resource may be about several entirely different topics. Semantic relatedness can be leveraged to reduce concept drift caused by multi-facetedness of entities. Connected entities can be attenuated depending on the semantic relatedness to a tag the surfer originated from.

3.1 AspectScore and InteliScore

AspectScore and InteliScore are inspired by the intelligent surfer model of PageRank, introduced in [21]. They, hence, dynamically adapt the graph representation of the folksonomy, depending on the set of query entities. The adaptation of the graph representation is based on tag types for AspectScore and on semantic relatedness for InteliScore. Further, they make use of a second graph-based ranking algorithm. In the following, the term *scorer* will refer to this second graph-based ranking algorithm. Both algorithms aim to alleviate the problem of concept drift of scorer. AspectScore and InteliScore are computed as follows, where for AspectScore the folksonomy extended by tag types (Definiton 1) is used. Hence, in the first step, the tags in the folksonomy are disambiguated. For InteliScore, the folksonomy definition in [12] is used.

1. The query entities are transformed into a set of query tags. E.g for a ranking task *interests match*, the user query entity is transformed into a set of query tags. Therefore, the tags are weighted by the usage frequency of the user. For a folksonomy extended by tag types, a parameter δ_{type} acts as an additionally weighting factor depending on a tag's type.
2. For each query tag, a folksonomy graph representation $G = (V, E)$, as used by scorer, is created.
3. $G = (V, E)$ is adapted depending on the query tag. This adaptation is shown in Algorithm 3.1, where $\xi = U \cup T \cup R$ is the set of entities in the folksonomy.

Algorithm 3.1 Edge weight adaptation according to tag types or semantic relatedness

Input: Edges E, Query tag $t_q \in T$
procedure AdaptEdgeWeights(E, t_q)
  for all $(e \to t, w) \in E, e \in \xi, t \in T, w \in \mathbb{R}$ **do**
    $w = \text{AdaptEdgeWeight}(w, t_q, t)$
  for all $(e \to u, w) \in E, e \in \xi, u \in U, w \in \mathbb{R}$ **do**
    $TE = \{(u \to t, w_t) | (u \to t, w_t) \in E)\}$
    $w = \text{AdaptEdgeWeight}(w, t_q, \text{Summarize}(TE))$
  for all $(e \to r, w) \in E, e \in \xi, r \in R, w \in \mathbb{R}$ **do**
    $TE = \{(r \to t, w_t) | (r \to t, w_t) \in E)\}$
    $w = \text{AdaptEdgeWeight}(w, t_q, \text{Summarize}(TE))$
Output: Edges E
Input: Weight $w \in \mathbb{R}$, Tags $t_q, t \in T$
procedure AdaptEdgeWeight(w, t_q, t)
  if AspectScore **then**
    $w_{adapted} = \gamma_{t_q.type, t.type} \cdot w$;
  if InteliScore **then**
    $w_{adapted} = \text{SemanticRelatedness}(t, t_q) \cdot w$
Output: Weight $w_{adapted}$

4. The actual ranking of entities is performed by scorer.
5. The results for each query entity are accumulated using the weights of the query tags.

4. EVALUATION

The evaluation comprises of the resource ranking tasks *interests match* and *guided search*. Results from *guided search* for a set of evaluation methodologies are presented in detail while the remainder of the results is summarized.

4.1 Evaluation Methodology

LeavePostOut is introduced in [14] for the task of tag recommendation in folksonomies. A post $P_{u,r}$ is composed of all tag assignments by user u to resource r and P is the set of all posts. LeavePostOut removes one post at a time from the folksonomy. A subset of the entities part of the post are used as query entities to create a ranking. The recommendation algorithm comes up with the tags t that appear in tag assignments of $P_{u,r}$ given u and r as query entities [14]. To use the methodology for the task of resource ranking, resource r of $P_{u,r}$ has to be ranked at the top or as high as possible. Therewith the Assumption 2 is made:

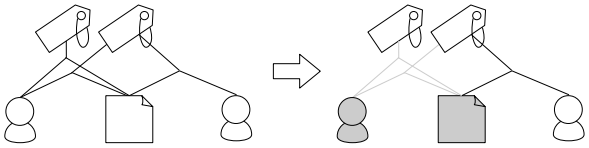

Figure 3: LeavePostOut methodology

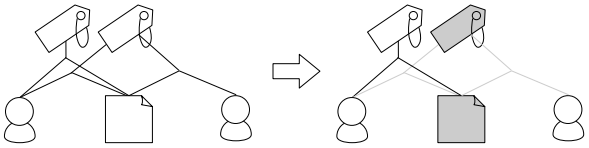

Figure 4: LeaveRTOut methodology

Assumption 2. The assignment of a tag by a user to a resource indicates relevance of the resource towards the information need represented by the assigned tag, and represented by the user.

Additionally, as the task of resource ranking requires the assessment of relevance of each resource in the ranking, but with LeavePostOut the ranking's quality can only be assessed with regard to the relevance of resource r, the assessment of the ranking's overall quality is limited. For example, given a ranking, it is only known that r is of relevance towards the query. However, this does not mean that none of the other highly ranked resources are not relevant. This problem is described as the incompleteness problem in [9]. The key observation of the LeavePostOut methodology is, that after removal of $P_{u,r}$, there is no information in the folksonomy anymore, that connects the user u of $P_{u,r}$ directly with resource r. However, there remains the possibility, that information in the folksonomy still exists which connects resource r or user u directly to the tags of tag assignments in $P_{u,r}$. This is illustrated in Figure 3.

Hence, the methodology provides a substantially harder problem for the task *interests match* than *guided search*. This is because for the task *interests match*, u is used as query entity, which in the folksonomy, is no longer related to r. *Guided search*, however uses a subset of the tags of the post as query entity, which are potentially still connected to r. It is additionally possible to evaluate the task of combinations of the two. The task *more like this*, however, can not be evaluated with this methodology, as a post contains only one resource and this resource r cannot act as query entity and scored entity at the same time. Carmel et al. point out, that to overcome the incompleteness problem, results obtained from a LeavePostOut evaluation should be validated with alternative evaluation methodologies [9]. Therefore, for this work, LeavePostOut is complemented with *LeaveNPostsOut*, *LeaveRTOut* and *LeaveNRTsOut*. A possibility to alleviate the incompleteness problem for the task *interests match* is to use the variation LeaveNPostsOut, which, instead of removing one post $P_{u,r}$, removes n random posts. Hence, $\frac{|P|}{n}$ posts of each user u are taken out on average. The ranking algorithm, then, has to rank resource r of any removed post $P_{u,r}$ of user u, for *interests match*, at the top of the ranking, or as high as possible. For *guided search*, using t as query entity, the ranking algorithm has to rank resource r of any removed post $P_{u,r}$, in which tag t appears in a tag assignment, at the top of the ranking, or as high as possible. This methodology allows for a trade-off between how much data of a corpus can be used as information to create a ranking, and alleviating the incompleteness problem.

The proposed LeaveRTOut evaluation methodology is inspired by the key observation made from LeavePostOut. In LeavePostOut, after $P_{u,r}$ is removed, u and r are considered unconnected. But a tag t in a tag assignment of $P_{u,r}$ may still be connected to r. An alternative is thus the proposed LeaveRTOut methodology, which instead of eliminating the connection in the folksonomy between a user u and a resource r, eliminates the connection in the folksonomy between a tag t and a resource r as illustrated in Figure 4 and described in Algorithm 4.1. Similar to LeavePostOut, LeaveRTOut makes Assumption 2. In contrast to LeavePostOut, in LeaveRTOut, the task *guided search* is substantially harder to solve than *interests match*. In addition, it is possible to evaluate the task combinations of the two. The task *more like this*, is however not possible to evaluate for the same reasons as for LeavePostOut. Similar to LeaveNPostsOut, LeaveNRTsOut can be used to alleviate the incompleteness problem. LeaveNRTsOut can alleviate the incompleteness problem for the task *guided search*.

Algorithm 4.1 LeaveRTOut evaluation methodology

Input: Folksonomy $F = (U, R, T, Y)$
1: **procedure** LeaveRTOut(F)
2: **for all** $t \in T$ **do**
3: **for all** $r \in R$, where $\exists(u, r, t) \in Y$ **do**
4: $RT = \emptyset$
5: **for all** $(u, r, t) \in Y$ **do**
6: $F = F \backslash (u, r, t)$
7: $RT = RT \cup (u, r, t)$
8: AssessRankingQuality(Score(F,
9: CreateQueryEntities(RT)))
10: **for all** $(u, r, t) \in RT$ **do**
11: $F = F \cup (u, r, t)$

The metrics *Mean Average Precision (MAP)*, *Average Precision (AP)* [17] and *Mean Normalized Precision (MNP) at k* are used in the evaluation. *MNP at k* is derived from *Precision at k* [17] 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. respect the maximal achievable $Precision_{max}(k)$.

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

4.2 Corpus

As folksonomy corpus, a dump [16] of the publication management system BibSonomy is used. BibSonomy allows to tag scientific publications (bibtex resources) and arbitrary resources addressable via a URL (bookmark resources). A p-core [14] of level l guarantees a corpus to contain only entities that appear in at least l posts. In this work, $l = 5$ is used to extract a p-core of the corpus and hence reduce noise due to e.g. infrequent tags, and to focus on the dense part of the folksonomy, e.g. frequent users. Before a p-core is extracted, the corpus is reduced to a manageable size. Hence, tag assignments for both bookmarks and bibtex resources are added iteratively in temporal order, beginning with the oldest, until a manageable size for evaluation has been obtained. Hence, before the extraction of a p-core, the reduced corpus consists of as many bookmarks as bibtex tag assignments. The characteristics of the corpus before and after

the extraction of a p-core are shown in Table 1.

To enhance the corpus with tag types, the tag assignments in the corpus are manually labeled with tag types. The resulting distribution of tag types is given in Table 2.

4.3 Parameterization

LeavePostOut is used to determine parameter values for *interests match*. The ranking effectiveness is measured with MAP. FolkRank is parameterized with the biased jump probability α of the biased surfer model. In a sensitivity analysis, $\alpha = 0.05$ is found to be most effective. AspectScore is parameterized with the implementation of scorer. FolkRank is used as the implementation of scorer. SUMMARIZE($tagEdges$) is implemented such, that a maximal $w_{adapted}$ results from ADAPTEDGEWEIGHT(w, t_{type_q}, SUMMARIZE($tagEdges$)). In the following, the analysis of useful values for parameters δ_{type} and $\gamma_{type_q,type}$ of AspectScore is described. The *Location* tag type is neglected as it is not contained in the corpus. Recall, that the parameter evaluation is conducted with the user of a post as query entity. As AspectScore transforms query entities to query tags, the *first analysis* investigates the influence of a single tag type within all tag types the user query entity may be transformed into. Hence, values for δ_{type} and $\gamma_{type_q,type}$ are analyzed and set as follows: δ_{type_A} is varied in steps of 0.1 from 0.0 to 1.0. $\delta_{type_B} = 1.0, \forall type_B \neq type_A$. $\gamma_{type_q,type} = 1$ if $type_q = type$ and $\gamma_{type_q,type} = 0$ otherwise. With this setting of $\gamma_{type_q,type}$, FolkRank's surfer is limited to paths that only pass nodes of tags of a certain tag type. However, paths that only pass user and resource nodes, and do not pass a tag node at all are possible. To reduce computation time, in sum, 33% of posts are left out at each measuring point. Figure 5 presents the results for the setups described. As can be

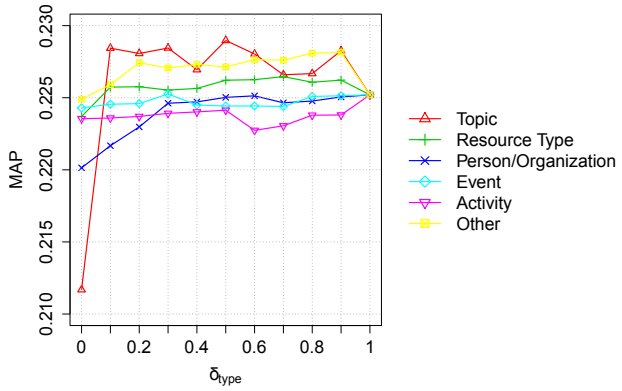

Figure 5: MAP for first analysis of AspectScore

seen, the influence of none of the tag types is as distinct as the tag type *Topic*. However, the inclusion of any of the tag types is beneficial to some extent, as MAP rises e.g. from

Table 1: Corpus characteristics before and after preparation

	Before	After
Users	7243	69
Bookmark resources	281550	9
Bibtex resources	469654	134
Tags	216094	179
Tag assignments	2740834	3269
Bookmark posts	330192	51
Bibtex posts	526691	959

Table 2: Tag type distribution in corpus

Topic	Other	Resource Type	Event	Person/ Organization	Activity	Location
2225	486	198	182	143	35	0

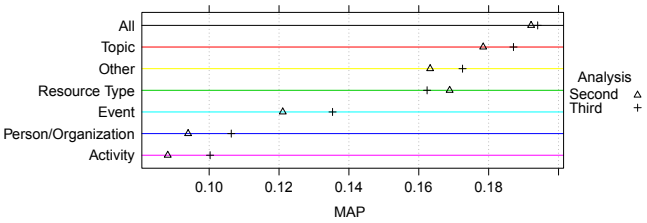

Figure 6: MAP for second and third analysis of AspectScore

$\delta_{type} = 0.0$ to $\delta_{type} = 0.1$. Hence, it can be concluded, that, in this evaluation setup, any of the tag types is useful for ranking. As shown in the first analysis, any type, except *Topic* does not appear to have a very distinct impact on the effectiveness of the ranking when used in combination with all other tag types. Hence, in the *second analysis*, the goal is to investigate the effectiveness of a ranking by using tags of certain types in isolation. Therefore $\gamma_{type_q,type}$ is set as described in the first analysis and δ_{type} is set as follows: $\delta_{type} = 1$ if *type* describes the tag type of the tags a user is transformed into is to be limited to and $\delta_{type} = 0$ otherwise. In a *third analysis*, δ_{type} is set as described in the second analysis. However, $\gamma_{type_q,type} = 1$ in any case in this analysis. Hence, in this analysis, the surfer is no longer limited to paths that pass only nodes of tags of a certain tag type. Figure 6 presents the MAP for the setups of the second and third analysis. For both analyses, 100% of the corpus' posts are used. As can be seen, allowing the user to be transformed into all tag types is most effective. Additionally, not limiting the path to any of the tag types, but instead allowing all tag types in a path, is superior for all tag types but *Resource Type*. Hence, in a *fourth analysis*, similar to the first analysis, the influence of a single tag type within all tag types the user may be transformed into is analyzed. The setup is equal to the one of the first analysis, with the difference of setting $\gamma_{type_q,type} = 1$ for all $(type_q, type)$ combinations combinations. Figure 7 presents the results for this setup. As can be seen, no improvement in MAP can be achieved by limiting the influence of tags the user may be transformed into to a certain tag type. For the results of this parameter sensitivity analysis, in the evaluation, the parameters of AspectScore are set as follows: $\delta_{type} = 1$ and $\gamma_{type_q,type} = 1$. Hence, AspectScore is reduced to FolkRank on a by tag types disambiguated graph of the folksonomy. The results of this section are somewhat contrary to the

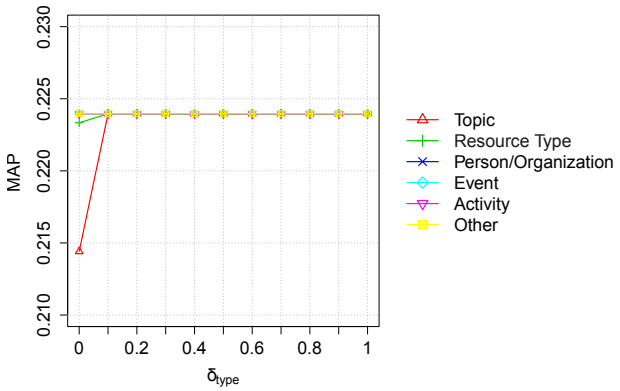

Figure 7: MAP for fourth analysis of AspectScore

Table 3: MAP for LeavePostOut and *guided search*

Popularity	FolkRank	AspectScore	InteliScore
0.0937	0.2136	0.2240	0.1801

results obtained by Cantador et al. in [8]. Cantador et al. show that some categories of tags are more useful for ranking than others. They argue that by not considering certain categories of tags, noise may be reduced. However, in their work, no p-core is taken, which also reduces noise in a corpus. InteliScore is parameterized with the implementation of scorer. FolkRank is used as the implementation of scorer. The implementation of SEMANTICRELATEDNESS(t, t_q) is another parameter. For the evaluation, XESA [22] based on the English Wikipedia is used. The following linguistic pre-processing steps are performed: (i) Tokenization based on whitespace and punctuation. (ii) Relatedness of a tag consisting of multiple tokens is averaged. (iii) Lowercase normalization and (iv) Stemming. As XESA is based on Wikipedia, the semantic relatedness between pairs of tags, which include about 27% of the tags in the corpus, cannot be determined. This may be due to the fact, that terms of one of the two tags are considered as stopwords, numbers e.g. *2006*, or they may not appear in Wikipedia frequently enough, e.g. *itegpub*. In these cases, the semantic relatedness is taken as 0.0. SUMMARIZE($tagEdges$) is implemented such, that a maximal $w_{adapted}$ results from ADAPTEDGEWEIGHT(w, t_{type_q}, SUMMARIZE($tagEdges$)).

4.4 Results

To compare the results with a simple ranking algorithm, a popularity ranking is added. Popularity is simply computed for a resource r as the sum of the number of tags assigned to the resource r and the number of users that used r in a tag assignment. The score of a resource is therefore query-independent.

Significance tests are conducted to determine statistical significance of effectiveness of the overall ranking of the algorithms. These tests are based on the AP, which measures the overall ranking effectiveness achieved for an information need. As the AP does not follow a normal distribution, but can be compared pairwise, *Wilcoxon signed-rank tests*[1] are conducted. One exception from this, however, are comparisons with AspectScore. As AspectScore disambiguates tags, the number of created rankings varies from other algorithms for the tasks of *guided search* regardless of the evaluation methodology, and for the task of *interests match* evaluated with LeaveRTOut, or LeaveNRTsOut. In these cases, the comparison cannot be made pairwise and therefore *Wilcoxon rank-sum tests*[2] are conducted. The null hypothesis H_0 is that a pair of compared algorithms yields identical effectiveness, as measured by AP. H_1 states, that one of the challenging algorithms is more effective than its contester. $p = 0.05$ is used as significance level. The results of all pairwise comparisons are shown in the respective sections in the following. In the following, LeavePostOut is used for the task *guided search*. Each of the tags of the left out post is once used as single query entity to create a ranking. Figure 8 shows the results of positions where relevant resources are found as a violin plot [11]. Table 3 and Figure 9 show the results of the metrics MAP and MNP at k for $k \in [1, 10]$ respectively. As can be seen, AspectScore

[1]http://stat.ethz.ch/R-manual/R-patched/library/stats/html/wilcox.test.html, retrieved 19/03/12

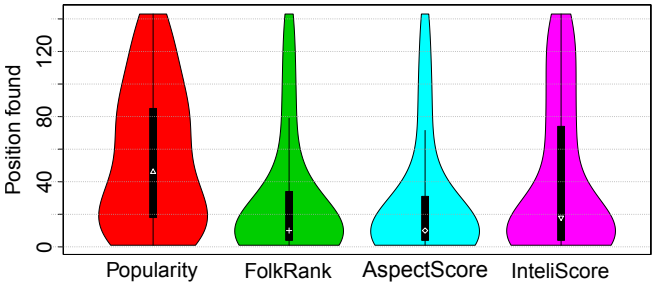

Figure 8: Violinplot for LeavePostOut and *guided search*

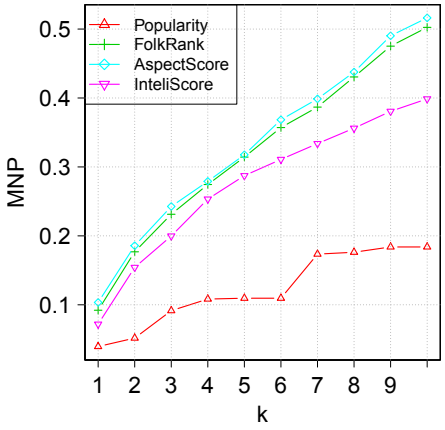

Figure 9: MNP at k for LeavePostOut and *guided search*

performs slightly better than FolkRank with a MAP of approximately 0.22. InteliScore (0.18) is only better than Popularity (0.09) in MAP. Similarly, the algorithms are effective with ranking in the top positions. The results show that Popularity is not sufficient for ranking in this task. Moreover, disambiguation in AspectScore shows an improvement over FolkRank in MAP, while it cannot be said to be significantly more effective. InteliScore performs significantly worse than FolkRank and AspectScore. A reason for this is the difficulty to determine the semantic relatedness of tags. The results of all pairwise comparisons for statistical significance are shown in Table 4. In the following, LeaveRTOut is used for the task *guided search*. The tag of the left out connection between a resource and tag is hence used as a single query entity. Figure 10 shows the results of positions where relevant resources are found as a violin plot. Table 5 and Figure 11 show the results of the metrics MAP and MNP at k for $k \in [1, 10]$ respectively. As can be seen, Popularity achieves the highest MAP (0.08), even though in Figure 10, the median is worse compared to e.g. FolkRank. This is due to the good ranking performance in the very top positions which can be seen in Figure 11. With regard to

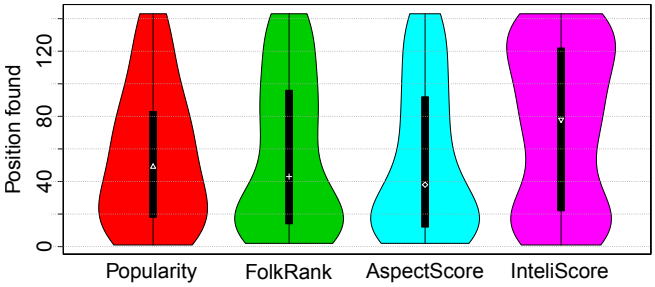

Figure 10: Violinplot for LeaveRTOut and *guided search*

Table 4: Significance comparisons for LeavePostOut and *guided search*

More effective than →	Popularity	FolkRank	AspectScore	InteliScore
Popularity	□	□	□	□
FolkRank	⊠	□	□	⊠
AspectScore	⊠	□	□	⊠
InteliScore	⊠	□	□	□

Table 6: Significance comparisons for LeaveRTOut and *guided search*

More effective than →	Popularity	FolkRank	AspectScore	InteliScore
Popularity	□	□	□	⊠
FolkRank	⊠	□	□	⊠
AspectScore	⊠	⊠	□	⊠
InteliScore	□	□	□	□

Table 5: MAP for LeaveRTOut and *guided search*

Popularity	FolkRank	AspectScore	InteliScore
0.0834	0.0529	0.0589	0.0433

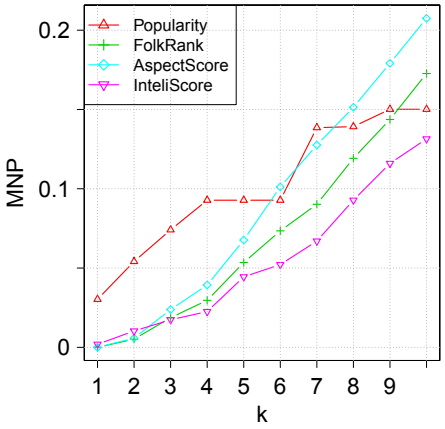

Figure 11: MNP at k for LeaveRTOut and *guided search*

the overall ranking effectiveness, AspectScore (0.06) is more effective than FolkRank (0.05). InteliScore is worst with regard to MAP (0.04). With regard to ranking in the top positions Popularity is only outclassed at $k = 6$ and $k \in [8, 10]$. Beside Popularity, the effectiveness of ranking in the top positions is mostly in the order of AspectScore, FolkRank, and InteliScore being worst. The results show that Popularity in fact performs best for ranking in the very top positions in this task. Moreover, disambiguation in AspectScore again shows an improvement over FolkRank in MAP. In this evaluation setup AspectScore is significantly more effective than FolkRank. InteliScore performs worst for reasons given previously. The results of all pairwise comparisons for statistical significance are shown in Table 6. The different results of Popularity may be explained by the fact, that as described in Section 4.1, after the removal of a post, the connection between the tag and the resource to be found may still exist. As in the task *guided search*, the tag is used as query entity, it is a substantially easier task for the graph-based algorithms to find a relevant resource that is still directly connected to the query entity. Popularity does not make use of these connections and hence performs worse.

4.5 Synopsis

Table 7 summarizes the findings of the statistical significance tests for all evaluation setups conducted. For the respective tasks and evaluation methodologies the algorithms that win most pairwise statistical significance comparisons are shown. For LeaveNPostsOut and LeaveNRTsOut $\frac{|P|}{n} = \frac{1}{3}$ and $\frac{|RT|}{n} = \frac{1}{3}$ is used respectively. For *interests match*, the results between LeavePostOut or LeaveNPostsOut differ from the results obtained with LeaveRTOut or LeaveNRTsOut. This is due to the fact, that they set a differently hard task to solve. The results from LeavePostOut and LeaveNPostsOut are more useful to assess the effectiveness for *interests match* in a resource recommendation task. There, no connection between the user and a potential relevant resource exists. LeaveRTOut and LeaveNRTsOut are more useful to assess the effectiveness for *interests match*, in which e.g. the current resources are to be presented in order of how they match the user's interest. For *guided search*, and with regard to significance comparisons of overall ranking effectiveness, the results between LeavePostOut or LeaveNPostsOut do not differ from the results obtained with LeaveRTOut or LeaveNRTsOut. However, in general, the results obtained from LeaveRTOut and LeaveNRTsOut are more useful to assess the effectiveness for *guided search* in a scenario in which no connection between the tag searched for, and a potential relevant resource can be expected. LeavePostOut and LeaveNPostsOut are more useful to assess the effectiveness for *guided search*, in which such a connection can be expected.

Table 7: Summary of significance comparisons

Methodology	*Interests match*	*Guided search*
LeavePostOut	AspectScore	FolkRank, AspectScore
LeaveNPostsOut	AspectScore	FolkRank
LeaveRTOut	FolkRank, AspectScore, InteliScore	AspectScore
LeaveNRTsOut	FolkRank, InteliScore	FolkRank

InteliScore did not perform well for two reasons. Firstly, determining the semantic relatedness of tags poses a great challenge. Secondly, for *interests match*, the user query entity has to be transformed into tag query entities, thereby, valuable information offered by resources connected to the user is lost. Disambiguation of tags, however, has shown to be helpful for AspectScore. To obtain these results, the parameterization was done using an analysis of MAP results obtained from LeavePostOut and the task *interests match*. Hence, the algorithms may perform better with regard to a metric, or task if parameterized accordingly. Additionally, the statistical significance is computed based on AP, 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.

5. CONCLUSION

In this paper, AspectScore and InteliScore are proposed that extend graph-based ranking algorithms in folksonomies by dynamically adapting the graph representation depending on a given query entity. However, both require semantic information which is usually not found in folksonomies. Limitations of this work lie in the dataset which was labeled manually as no dataset with tag types that is sufficient for evaluation is presently available. Manual labelling is cumbersome, as only the user who assigns a tag actually knows the true tag type. Additionally, folksonomy applications impact the nature, type and role of tags [18]. A corpus of an e-learning application having tag types like CROKODIL[2] [3] for example, may therefore have different characteristics. Hence it is of interest to evaluate these approaches with such a corpus in future. Furthermore, in future, the determination of semantic relatedness of tags and their linguistic pre-processing steps may be reconsidered. In this work, XESA based on the English Wikipedia was used and for a significant part of the tags in the corpus, no semantic relatedness could be determined. It may therefore be beneficial to leverage other semantic relatedness measures as folksonomies often contain tags, which are not widely used in Wikipedia. Such a measure could additionally be combined with the knowledge of tag types. Depending on the application scenario, it may even be useful to determine the relatedness of tags of type *Location* based on their distance in the real world. In addition, sentiment analysis on tags of tag type *Other* could be performed and the folksonomy graph adapted accordingly. Moreover, ambiguity in folksonomies on the level of linguistic knowledge of semantics, e.g. using context-specific information could be investigated further. Finally, a user-study may determine the true utility of ranking for users as the relevance assumption in this work may not be applicable to all ranking scenarios.

6. REFERENCES

[1] F. Abel. *Contextualization, User Modeling and Personalization in the Social Web*. PhD Thesis, Gottfried Wilhelm Leibniz Universität Hannover, 2011.

[2] M. Ames and M. Naaman. Why We Tag: Motivations for Annotation in Mobile and Online Media. In *Proc. of the SIGCHI*, pages 971–980, 2007.

[3] M. Anjorin, C. Rensing, K. Bischoff, C. Bogner, L. Lehmann, A. Reger, N. Faltin, A. Steinacker, A. Lüdemann, and R. Domínguez García. CROKODIL - A Platform for Collaborative Resource-Based Learning. In *Proc. of the 6th EC-TEL*, pages 29–42, 2011.

[4] C. Au Yeung, N. Gibbins, and N. Shadbolt. Contextualising Tags in Collaborative Tagging Systems. In *Proc. of the 20th ACM HyperText*, pages 251–260, 2009.

[5] T. Bogers. *Recommender Systems for Social Bookmarking*. PhD Thesis, Tilburg University, 2009.

[6] D. Böhnstedt, P. Scholl, C. Rensing, and R. Steinmetz. Collaborative Semantic Tagging of Web Resources on the Basis of Individual Knowledge Networks. In *Proc. of the 17th UMAP*, pages 379–384, 2009.

[7] S. Brin and L. Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. In *Proc. of the 7th WWW*, pages 107–117, 1998.

[8] I. Cantador, I. Konstas, and J. Jose. Categorising Social Tags to Improve Folksonomy-Based Recommendations. *Web Semantics: Science, Services and Agents on the World Wide Web*, 9:1–15, 2011.

[9] D. Carmel, N. Zwerdling, I. Guy, S. Ofek-Koifman, N. Har'el, I. Ronen, E. Uziel, S. Yogev, and S. Chernov. Personalized Social Search Based on the User's Social Network. In *Proc. of the 18th ACM CIKM*, pages 1227–1236, 2009.

[10] T. Haveliwala. Topic-Sensitive PageRank. In *Proc. of the 11th WWW*, 2002.

[11] J. Hintze and R. Nelson. Violin plots: A box plot-density trace synergism. *The American Statistician*, 52(2):181–184, 1998.

[12] A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme. BibSonomy: A Social Bookmark and Publication Sharing System. In *Proc. of the Conceptual Structures Tool Interoperability Workshop*, 2006.

[13] A. Hotho, R. Jäschke, C. Schmitz, and G. Stumme. Information Retrieval in Folksonomies: Search and Ranking. In *Proc. of the 3rd ESWC*, pages 411–426, 2006.

[14] R. Jäschke, L. Marinho, A. Hotho, S.-T. L., and S. G. Tag Recommendations in Folksonomies. In *Proc. of the 11th PKDD*, pages 506–514, 2007.

[15] D. Jurafsky and J. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, 2008.

[16] Knowledge and D. E. Group. University of Kassel: Benchmark Folksonomy Data from BibSonomy. http://www.kde.cs.uni-kassel.de/bibsonomy/dumps/2011-07-01.tgz, July 2011. Retrieved 11/10/11.

[17] C. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. CUP, 2008.

[18] C. Marlow, M. Naaman, D. Boyd, and M. Davis. HT06, Tagging Paper, Taxonomy, Flickr, Academic Article, To Read. In *Proc. of the 7th ACM HyperText*, pages 31–40, 2006.

[19] I. Peters. *Folksonomies: Indexing and Retrieval in Web 2.0*. De Gruyter Saur, 2010.

[20] M. Ramezani. *Using Data Mining for Facilitating User Contributions in the Social Semantic Web*. PhD Thesis, Karlsruher Institut für Technologie, 2011.

[21] M. Richardson and P. Domingos. The Intelligent Surfer: Probabilistic Combination of Link and Content Information in PageRank. *NIPS*, 14:1441–1448, 2002.

[22] P. Scholl, D. Böhnstedt, R. Domínguez García, C. Rensing, and R. Steinmetz. Extended Explicit Semantic Analysis for Calculating Semantic Relatedness of Web Resources. In *Proc. of the 5th EC-TEL*, pages 324–339, 2010.

[23] J. Surowiecki. *The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations*. Doubleday, 2004.

[2] http://www.crokodil.de/, retrieved 02/20/2012