Folksonomy-based Collabulary Learning

Leandro Balby Marinho, Krisztian Buza and Lars Schmidt-Thieme

Information Systems and Machine Learning Lab (ISMLL)
Samelsonplatz 1, University of Hildesheim, D-31141 Hildesheim, Germany
{marinho,buza,schmidt-thieme}@ismll.uni-hildesheim.de

Abstract. The growing popularity of social tagging systems promises to alleviate the knowledge bottleneck that slows down the full materialization of the Semantic Web since these systems allow ordinary users to create and share knowledge in a simple, cheap, and scalable representation, usually known as folksonomy. However, for the sake of knowledge workflow, one needs to find a compromise between the uncontrolled nature of folksonomies and the controlled and more systematic vocabulary of domain experts. In this paper we propose to address this concern by devising a method that automatically enriches a folksonomy with domain expert knowledge and by introducing a novel algorithm based on frequent itemset mining techniques to efficiently learn an ontology over the enriched folksonomy. In order to quantitatively assess our method, we propose a new benchmark for task-based ontology evaluation where the quality of the ontologies is measured based on how helpful they are for the task of personalized information finding. We conduct experiments on real data and empirically show the effectiveness of our approach.

1 Introduction

Due to the concrete advances towards the Semantic Web vision [4], ontologies are growing in use, specially in areas concerning information finding and organization. However, their massive adoption is severely shortened because of the effort one needs to take to assemble them, task which is usually assigned to domain experts and knowledge engineers. Although ontology learning can help to some extent, the participation of the expert is still usually required since the learned representations are not free of inconsistences (in a semantic level at least) and therefore require manual validation and fine tuning. A more promising solution to this problem lies in the rapid spread of the Web 2.0 paradigm as it has the potential to educate ordinary users towards voluntary semantic annotation, thereby decentralizing and cheapening knowledge acquisition. The increasing popularity of Web 2.0 applications can be partly explained by the fact that no specific skills are needed for participating, where anyone is free to add and categorize resources in the form of free keywords called *tags*. Tags do not need to conform to a closed vocabulary and therefore reflect the latest terminology in the domain under which the system operates. Furthermore, the exposure to each other tags and resources creates a fundamental trigger for communication and sharing, thus lowering the barriers to cooperation and contributing to the creation of collaborative lightweight knowledge structures known as *folksonomies*. Despite the compelling idea of folksonomies, its uncontrolled nature can bring problems, such as: synonymy, homonymy, and polysemy, which lowers the efficiency of content indexing and searching. Another problem is that folksonomies usually disregard relations between their tags, what restricts the support for content retrieval. If tags are informally defined and continually changing, then it becomes difficult to automate knowledge workflow. In this sense, it is necessary to find a compromise between the flexibility and dynamics of folksonomies and the more systematic structure of controlled vocabularies. This compromise is usually known as *collabulary* [1], which corresponds to a portmanteau of the words *collaborative* and *vocabulary*. For our purposes we define a collabulary in terms of a special ontology that represents the knowledge of both users and experts in an integrated fashion.

In this paper we propose a method for collabulary learning. To this end, we first take a folksonomy and a domain-expert ontology as input and project them into an enriched folksonomy through semantic mapping; we then apply a fast and flexible algorithm based on frequent itemset techniques to learn an ontology over the enriched folksonomy. The main contributions of this paper are: (i) a definition for the new problem of *collabulary learning*, (ii) a method for automatically enriching folksonomies with domain-expert knowledge, (iii) a fast and flexible algorithm based on efficient frequent itemset mining techniques for ontology learning from folksonomies, and (iv) a new benchmark for task-based ontology evaluation in folksonomies.

An obvious question one could ask is to which extent this so called *collabulary* really helps. Looking at the literature on ontology learning from folksonomies (e. g., [26, 15, 9, 23, 22]) we see that most of the proposed approaches are motivated by facilitating navigation and information finding, even though they do not quantify to which extent the learned ontologies really help on this task. Instead, the quality of the learned ontologies is measured based on how good they match people's common sense or how similar they are to a reference ontology. We argue that in this context, an ontology is as good as it helps users finding useful information. Therefore, we propose, as contribution (iv), to plug the investigated knowledge structures in collaborative filtering algorithms for recommender systems and evaluate the outcome as an indicator of the ontologies' usefulness, given that collaborative filtering [21] is one of the most successful and prominent approaches for personalized information finding. To the best of our knowledge this is the first effort towards thorough empirical investigation of the tradeoff between folksonomies and controlled vocabularies. We conduct experiments on a real-life dataset and demonstrate the effectiveness of our approach.

The paper is organized as follows. Section 2 presents a definition for the collabulary learning problem and our approach for enriching a folksonomy with domain-expert vocabulary. Section 3 introduces an algorithm based on fast frequent itemset mining techniques for ontology learning from folksonomies. Section 4 presents a new benchmark for task-based ontology evaluation and Section 5 discusses the conducted experiments and their results, followed by related work and conclusions (Sections 6 and 7).

2 Folksonomy Enrichment

Our approach for folksonomy enrichment is based on providing a semantic mapping between an ontology designed by domain experts and a folksonomy, assuming that both describe the same domain over the same set of instances.

2.1 Problem definition

Before presenting our approach, we provide a simplified definition for some of the concepts used in this paper, namely, folksonomy, ontology, knowledge base and collabulary. Similarly to [16], we define a folksonomy as follows:

Definition 1. *A folksonomy*[1] *is a tuple* $\mathbb{F} := (U, T, R, Y)$ *where* U*,* T*, and* R *are finite sets, whose elements are called* users*,* tags *and* resources*, and* Y *is a ternary relation between them, i. e.,* $Y \subseteq U \times T \times R$*, whose elements are called tag assignments.*

Since taxonomies are central components of ontologies, we are going to focus on them first. Similarly to [12] we define an *ontology* as follows:

Definition 2. *An ontology is a tuple* $\mathcal{O} := (\mathcal{C}, root, \leq_{\mathcal{C}})$ *where* $\mathcal{C}$ *is a set of concept identifiers and* $\leq_{\mathcal{C}}$ *is partial order on* $\mathcal{C}$ *with one unique top element* root*, called taxonomy or concept hierarchy.*

A *knowledge base* in turn, is defined as follows:

Definition 3. *A knowledge base for an ontology* $\mathcal{O}$ *is a structure* $KB := (I, \iota_{\mathcal{C}})$ *where* I *is a set whose elements are instance identifiers and* $\iota_{\mathcal{C}} : \mathcal{C} \rightarrow 2^I$ *is a function associating concepts to instances called concept instantiation.*

To simplify our discussion, we assume that the relation between lexical terms and their associated concepts or instances is a bijection i. e., each lexical term is a identifier of a concept or an instance[2]. Finally we define the problem we want to address in this paper as follows:

Definition 4. *Given a folksonomy* $\mathbb{F}$*, an ontology* $\mathcal{O}$ *and a knowledge base* $KB_{\mathcal{O}} = (I, \iota_{\mathcal{C}})$ *for* $\mathcal{O}$*, an ontology* $\mathcal{P}$ *with concepts* $\mathcal{C}_{\mathcal{P}} = T_{\mathbb{F}} \dot\cup \mathcal{C}_{\mathcal{O}}$ *and a knowledge base* $KB_{\mathcal{P}} = (I', \iota'_{\mathcal{C}})$ *with* $I' = I \dot\cup R_{\mathbb{F}}$ *is called a* collabulary *over* $\mathbb{F}$ *and* $\mathcal{O}$*. The* ***collabulary learning problem*** *is here defined as finding a collabulary over* $\mathbb{F}$ *and* $\mathcal{O}$*, that best*[3] *represents the common knowledge between folksonomy users and domain experts.*

2.2 Semantic Mapping

Users of social tagging systems are heterogenous and thus have different levels of knowledge about a domain. Moreover, they can express very personal opinions about their resources, what lowers the potential for knowledge sharing. Tags like *stuff to chill*, *awesome artists*, or *makes me happy*[4] are very subjective and hence hard to make sense of, nevertheless they appear relatively often in real life folksonomies. In order to ensure interoperability, one needs to find a clear meaning for these tags such that we know, for example, that actually *stuff to chill* is related to *alternative*, *awesome artists* to *emo*

[1] In the original definition [16], it is additionally introduced a subtag/supertag relation, which we omit here since most of the real life folksonomies disregard such tag relations

[2] For our purposes instances will correspond to resources

[3] In this paper ”best” is defined in terms of an ontology-based application scenario

[4] Tags present in the online social radio station last.fm (http://last.fm)

and *makes me happy* to *rockabilly*, where *alternative*, *emo*, and *rockabilly* are concepts coming from a controlled and well agreed vocabulary[5]. We propose to address this issue by providing a semantic mapping between a folksonomy and a domain-expert ontology. As these two knowledge representations are structurally different (i. e., folksonomies do not have a partial order) we first turn the folksonomy, without loss of generality, into a *trivial ontology* $\mathcal{O}_{\mathbb{F}\downarrow T}$ over $\mathbb{F}$, i. e., a projection of the folksonomy to its tag space where all the concepts are leaf sibling nodes having root as father, more formally, $\forall t \in T : \leq_t := \emptyset$ and $\leq_{root} := T$ (e. g., Fig. 1.a). Now we can cast the semantic mapping, for our case, as an ontology matching problem, for which there is a well covered literature (e. g., [11, 13, 18]). Notice, however, that since the trivial ontology is structurally limited we can not rely on methods that heavily consider structural information.

Here we are interested in methods that depend only on the semantic content of the concepts involved since relying on syntactic descriptions of tags is not suitable, given the uncontrolled vocabulary of users. In [13] joint probability distributions are used as a framework for well-defined similarity measures that do not depend on the lexical layer of ontologies. Denoting the probability of two concepts A and B being identical by $P(A, B)$, in [13] it is shown that under certain general assumptions $P(A, B)$ for two concepts A and B coming from different ontologies, can be *approximated* as the fraction of instances that belong to both A and B[6], therefore reducing the problem to checking, for each instance, if it belongs to $A \cap B$. In our experiments we use the well known Jaccard coefficient

$$JS(A, B) := P(A \cap B)/P(A \cup B) := \frac{P(A, B)}{P(A, B) + P(A, \bar{B}) + P(\bar{A}, B)} \qquad (1)$$

as a representative of this family of similarity measures, where $P(A, \bar{B})$ is the probability that a randomly chosen instance belongs to A but not to B and $P(\bar{A}, B)$ the other way around. Having defined the similarity measure to use, we build the enriched folksonomy as follows:

1. Let $\mathcal{O}_D$ denote the domain-expert ontology and $\mathcal{O}_{\mathbb{F}}$ a trivial ontology, first we need to define a function for mapping the concept identifiers of both ontologies, i. e., $\hat{T} : \mathcal{C}_{\mathcal{O}_{\mathbb{F}}} \rightarrow \mathcal{C}_{\mathcal{O}_D}$, where $\mathcal{C}_{\mathcal{O}_{\mathbb{F}}}$ denotes the set of concept identifiers for $\mathcal{O}_{\mathbb{F}}$ and $\mathcal{C}_{\mathcal{O}_D}$ for $\mathcal{O}_D$. In our case, for each concept A in the *trivial ontology* representing the folksonomy, the most similar concepts from a *domain-expert ontology* is found as follows: $\hat{T}(A) := \underset{x \in \mathcal{C}_{\mathcal{O}_D}}{\operatorname{argmax}} JS(A, x)$ (see Example 1).
2. After that, we add the best mappings in $Y_{\mathbb{F}}$ as additional triples, i. e., $Y := Y \cup \{(u, \hat{T}(t), r) | (u, t, r) \in Y\}$.
3. Finally, we create a "dummy" user u_θ representing the expert and integrate it to the folksonomy. In other words, we build additional triples reflecting the concept instantiation of the expert and add them in $Y_{\mathbb{F}}$. The *enriched folksonomy* is now composed by the triples $Y := Y \cup \{(u_\theta, c, r) | c \in \mathcal{C}_{\mathcal{O}_D}, r \in \iota(c)\}$.

[5] With respect to music genres

[6] $P(A, B) \approx \frac{\text{count of instances belonging to both } A \text{ and } B}{\text{count of all instances}} = \frac{|\{\iota^*(A)\} \cup \{\iota^*(B)\}|}{\text{count of all instances}}$ where $\iota^*(C)$ denotes the extension of concept C, i. e., the set of resources belonging to C and its descendants

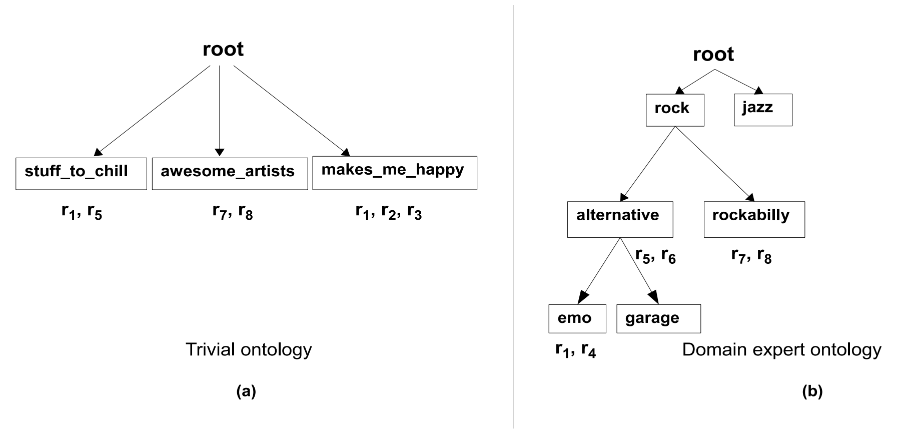


Fig. 1. Example of two knowledge bases associated with a trivial core ontology representing a folksonomy (a) and a domain-expert one (b) over the music domain.

Example 1. Consider the concept *stuff to chill* in the ontology at Fig. 1.a. Among the concepts in the ontology at Fig. 1.b, we have to find the one with the highest similarity. Computing a term like $P(\textit{stuff to chill}, \textit{alternative})$ is very simple as we just need to find the instances belonging to both concepts. Looking at Fig. 1, we observe that $\iota_C(\textit{stuff to chill}) := \{r1, r5\}$ and $\iota^*_C(\textit{alternative}) := \{r1, r4, r5, r6\}$ and therefore *stuff to chill* $\cap$ *alternative* $:= \{r1, r5\}$. Now $P(\textit{stuff to chill}, \textit{alternative})$ is just the number of elements in this intersection divided by the total number of distinct instances, i. e., $\frac{2}{8} := 0.25$. The same procedure can be repeated to find the terms $P(\overline{\textit{stuff to chill}}, \textit{alternative})$ and $P(\textit{stuff to chill}, \overline{\textit{alternative}})$. Now we just need to plug all these terms in Eq. 1. Given that *alternative* is indeed the best mapping, for the triples containing *stuff to chill*, i. e., $\{(u, r1, \textit{stuff to chill}), (u, r5, \textit{stuff to chill})\} \in Y$, we add new triples having *alternative*, i. e., $Y := Y \cup \{(u$,$r1$,*alternative*), (u,$r5$,*alternative*)$\}$. Finally the triples of the "expert user" are included, i. e., $Y := Y \cup \{(u_\theta$,$r5$, *alternative*), (u_θ,$r6$,*alternative*), (u_θ,$r7$,*rockabilly*), (u_θ,$r8$,*rockabilly*), (u_θ,$r1$,*emo*), (u_θ,$r4$,*emo*)$\}$.

Therefore, the enriched folksonomy has two major components: (i) a consistent vocabulary that properly matches the user vocabulary, and (ii) an additional user representing the expert point of view about the resources. The next step is to learn an ontology over this enriched folksonomy, which is the topic of the next section.

3 Frequent Itemsets for Learning Ontologies from Folksonomies

Most of the approaches concerning ontology learning from folksonomies rely on co-occurrence models (e. g., [26, 17, 9, 23, 22]). This is in line with the assumption that in sparse structures, such as folksonomies, positive correlations carry most of the essential information about the data (see [14] for a theoretical justification).

The idea of using frequent itemset mining for ontology learning in folksonomies is not new, in [22] for example, the authors conceptually proposed the exploitation of different projections of the folksonomy onto a two-dimensional formal context, where they applied association rule mining techniques. Our work is in line with this idea, however, we project the folksonomy to a transactional database[7], which is usual in the frequent itemset mining community and facilitates the direct application of highly efficient state-of-the-art methods. Furthermore, we explore additional assumptions on the users' resource tagging behaviour and how itemsets reflect relations between tags.

In this section, we propose a new algorithm for learning ontologies from folksonomies, which is based on frequent itemset mining on the one hand, and on extraction of taxonomic relationships from frequent itemsets on the other.

3.1 Frequent Itemset Mining

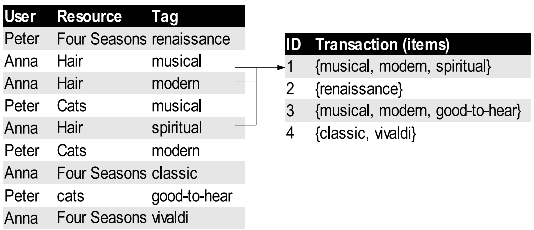
Searching for frequently co-occurring items is a well studied subject in data mining and is usually referred to as frequent itemset mining, for which there is a broad literature (e. g., [2, 3, 25, 19]). A frequent itemset is a set of frequently co-occurring items, like products often purchased together in a supermarket. The task can formally be defined as follows.

Definition 5. *Let $\Sigma = \{c_1, c_2, \dots, c_s\}$ be a set of items, a* transactional database *D is a subset of the power set of Σ, i. e., $D = \{C_1, C_2, \dots, C_n\}, \forall k \in \{1, 2, \dots, n\} : C_k \subseteq \Sigma$ where the sets $C_1, C_2, \dots, C_n$ denote co-occurring items and are called* transactions.

Definition 6. *The support for an* itemset *$I \subseteq \Sigma$, henceforth denoted as $sup(I)$, is defined as the number of supersets*[8] *of I in the transactional database $sup(I) = |\{C_k : C_k \in D, I \subseteq C_k, k \in \{1, 2, \dots, n\}\}|$*

Definition 7. *Given Σ, D and a minimum support threshold m, the* frequent itemset mining problem *is defined as the task of finding all sets $I \subseteq \Sigma$ with $sup(I) \geq m$.*

We project a folksonomy to a transactional dataset as follows. Given that Σ now corresponds to the set of tags, the transactions are identified by user-resource pairs where a transaction is composed by the tags used by the same user to the same resource (e. g., Fig. 2). Considering the transactional database illustrated in Fig. 2 and given that the minimum support threshold is set to 2, a frequent itemset would be $\{$*musical, modern* $\}$ for example.



User	Resource	Tag
Peter	Four Seasons	renaissance
Anna	Hair	musical
Anna	Hair	modern
Peter	Cats	musical
Anna	Hair	spiritual
Peter	Cats	modern
Anna	Four Seasons	classic
Peter	cats	good-to-hear
Anna	Four Seasons	vivaldi

ID	Transaction (items)
1	{musical, modern, spiritual}
2	{renaissance}
3	{musical, modern, good-to-hear}
4	{classic, vivaldi}

Fig. 2. Projection of a folksonomy to a transactional database

[7] see Definiton 5

[8] Note that for two subsets $C', I \subseteq \Sigma$, if $C' = I$, C' is a (trivial) superset of I.

3.2 Learning Ontologies from Folksonomies

Before introducing our method we define some intuitive **assumptions** used by the algorithm in the learning process:

1. **High Level Tag Assumption** – Users often associate resources with tags of different levels of an (eventually unknown) hierarchy. The more popular a tag is, the more general it is and therefore should occupy a higher level in the taxonomy to be learned. Notice that this in line with the assumption that users usually want to alleviate the cognitive effort by selecting tags representing broader concepts. Consider, for example, the well known singer *Elvis Presley* who is usually associated with the music genres *rock'n roll* and *rockabilly*, where *rock'n roll* is regarded as more general than *rockabilly*. If some users annotate this artist with *rock'n roll*, we expect that many of these users also use the tag *rockabilly*. However, as there are many other *rock'n roll* artists that are not necessarily *rockabilly*, we expect the tag *rock'n roll* to be used more often than *rockabilly*.
2. **Frequency Assumption** – If a frequent itemset F has significantly higher support than another frequent itemset F', we say that the items occuring in F are closer related to each other than the items in F', i.e. F should have more influence on the learned structure than F'.
3. **Large Itemset Assumption** – Suppose there are two frequent itemsets F_1 and F_2 and there is another frequent itemset F, $F \supseteq F_1 \cup F_2$. Suppose they all have approximatelly the same support. In this case (i. e., F, F_1, F_2 are frequent) we assume closer relation between the items included in $F_1 \cup F_2$, as if only F_1 and F_2 were frequent (but F not).

Algorithm 1 Algorithm for Taxonomy Learning based on Frequent Itemsets

```
Require: Folksonomy $data$, Array of Min. Support Tresholds[] $m$,
    Goodness Treshold $g$, Array of Edge Tresholds[] $e$,
    Resource-Tag-Correlation Treshold $c$.
Ensure: Learned Ontology $\mathcal{O}$ and its Knowledge Base $KB_{\mathcal{O}}$.
Transactional Database $D = projectData(data)$;
$S$=empty taxonomy;
for $i = 0; i <$m.$length; ++i$ do
    $fqSets = mineFrequentItemsets(D, m[i])$
    $S' = buildTaxonomyPieces(fqSets, e)$
    $S'' = prunePieces(S')$
    $S = addPrunedPiecesToTaxonomy(S, S'')$
end for
$\mathcal{O}, KB_{\mathcal{O}} = addResourcesToOntology(S, data, c)$
return $\mathcal{O}, KB_{\mathcal{O}}$
```

Our method is depicted in Algorithm 1. We apply an iterative process where the most frequent itemsets are mined first resulting in the learning of some pieces of the ontology. This corresponds to our *Frequency Assumption*, i. e., we first learn the relations contained in the most frequent itemsets since this corresponds to stronger evidences. Then in each subsequent iteration we relax the minimum support threshold in order to mine less frequent itemsets leading to the learning of new pieces. These pieces are iteratively put together converging to the final ontology. The main steps of the algorithm are detailed below.

For the frequent itemset mining (line 4), we use a highly efficient implementation of the algorithm Apriori [3, 2], which is based on a doubly recursive scheme to count how often a set of items co-occur (see [6, 7] for details). Since frequent itemsets do not always reflect the true relation between the items, we use the following *goodness condition* to which an itemset $I = \{t_1, t_2, \ldots, t_k\}$ needs to fulfill

$$\forall i, i \in \{1, 2, \ldots, k\} : \frac{\frac{sup(I)}{|D|}}{\frac{sup(\{t_i\})}{|D|} \cdot \frac{sup(I \backslash \{t_i\})}{|D|}} > g \tag{2}$$

where g is a *goodness threshold.*

After the mining step, we turn to the building of the taxonomy. We then start building what we call for convenience *taxonomy-pieces* (line 5), i. e., the taxonomic relations to be learned in the current iteration. In this graph, two nodes t_x and t_y are connected, with t_x being a superconcept of t_y, if there are frequent itemsets I containing both t_x and t_y such that $sup(\{t_x\}) \geq e[j] \cdot sup(\{t_y\})$, where $j = |I|$ and $e[j]$ is a given edge threshold for itemsets of size j. In other words, according to the *High Level Tag Assumption*, $sup(\{t_x\})$ has to be ”significantly larger” than $sup(\{t_y\})$. Moreover, due to the *Large Itemset Assumption* the co-occurrence of t_x and t_y in a large itemset means higher correlation between t_x and t_y than their co-occurrence in a smaller itemset, thus the meaning of ”significantly larger” depends on the size of the itemset I, i. e., the edge threshold $e[j]$ is the largest for 2-itemsets ($j = 2$), and the larger the itemset, the smaller the threshold.

To avoid multiple-inheritance relations, we prune the graph restricting it to a tree[9] (line 6). In this step we give preference to long paths since they are usually more informative. Given the edges $t_x \rightarrow t_y$, $t_y \rightarrow t_z$ and $t_x \rightarrow t_z$ (note that t_z has two father concepts, t_x and t_y) for example, the edge $t_x \rightarrow t_z$ would be considered redundant and thus would be removed, since the path going through t_y is longer[10]. This step is done by a depth-first-search like traversal through the *taxonomy-pieces*, which also guarantees that one concept has only one father-concept in the pruned graph.

As pointed out before, the relations learned in each iteration are merged with relations learned in previous iterations (line 7), hence converging to the final ontology. According to the *Frequency Assumption*, we ensure that itemsets with higher support have higher priority than the itemsets with lower support at the merging step, i. e., the taxonomy-pieces learned in previous iterations with higher support have higher priority than the ones learned in the current iteration (e. g., Fig. 3).

[9] The decision of using a tree instead of an arbitrary directed acyclic graph is due to the fact, that the evaluation procedure (see section 4) operates only on trees

[10] Note that the path $t_x \rightarrow t_y \rightarrow t_z$ leads to more specialized classification of the concept t_z, i. e., it is **not** classified as a subconcept of the general concept t_x, but it classified as a subconcept of a more specific concept t_y

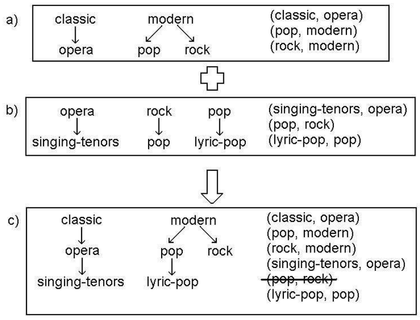


Fig. 3. Addition of taxonomical relations during the learning process. a) Relations already learned in previous iterations. b) Possible relations extracted in the current iteration. c) The learned taxonomy at the end of the current iteration. Note that only the new relations conforming to the old ones will be learned. In this example *pop* will not be defined as a subconcept of *rock*, because it is already known to be a subconcept of *modern* and a concept is only allowed to have one direct super-concept.

Remarks on complexity. The most expensive steps of the algorithm are: (i) the projection of the folksonomy to a transactional database, (ii) the extraction of frequent itemsets and (iii) the population of the ontology (line 9), as these are the only steps operating on the original folksonomy[11]. The other steps operate on the extracted frequent itemsets, which have sizes of lower orders of magnitude. The step (i) requires a linear scan on the folksonomy and therefore has linear complexity. As for the step (ii), the algorithm described in [3] can be implemented as l linear scans on the transactional database, where l is the size of the largest itemsets to be found. However, we implemented this step using a sophisticated trie representation of the transactional database and the frequent itemsets, which further improves efficiency [6, 7, 5]. The last step, i. e., the creation of the knowledge base (line 9, outside the loop), is based on the counting of the co-occurrence between resources and tags, which means a linear scan on the folksonomy.

4 Recommender Systems for Ontology Evaluation

The non-hierarchical property of folksonomies can somewhat restrict the capabilities of the users for finding information, as the browsing is constrained to a flat structure where tag relations are disregarded. Most of the literature concerning ontology learning from folksonomies use this observation as a main motivation for providing users with a taxonomy of tags, even though the authors do not quantify how good it is for the task it was designed for. Instead, the quality of the ontologies is measured based on whether they match people's common sense or a reference ontology. Taking this into account, and given that collaborative filtering [21] is one of the most successful and prominent approaches for helping users finding useful information, we propose to plug the inves-

[11] Or its projection, which have roughly the same size

tigated knowledge structures in recommender systems and evaluate the outcome as an indicator of their usefulness.

According to Porzel et al. [20], the minimal elements necessary for a task-based evaluation of an ontology are: *a task*, *one or more ontologies*, *an application* and a *gold standard*, which are specified for our benchmark as follows.

The Task: Recommending useful resources (e. g., music tracks, videos, websites, etc.) to the users, based on their implicit feedback on resources.

One (or more) Ontologies: A *trivial ontology* representing a folksonomy, a *domain-expert* ontology and a *collabulary*.

The Application: A collaborative filtering algorithm that uses taxonomies for finding the resources of interest. The idea of collaborative filtering is to suggest new resources based on the opinion of like minded users usually called *neighborhood*.

A Gold Standard: In this case the gold standard is the set of resources that the user prefers the most. In a typical recommender systems evaluation scenario we know these resources in advance. We split this set into training and test sets and then use the ones in the training set to recommend other resources the user would eventually like. To measure the quality of the recommender, we compare the predicted resources with the ones in the test set.

The concrete application we have chosen is a taxonomy-driven approach for recommender systems proposed by Ziegler et al. [27], where taxonomies are not only regarded as background knowledge, but also the main cornerstone for efficient and personalized information discovery. The algorithm heavily relies on the taxonomy to make the recommendations and therefore we assume that the better the taxonomy is, the better are the recommendations. In their method a user profile is not composed by vectors such as $\boldsymbol{u_i} \in \mathbb{R}^{|R|}$, where $\boldsymbol{u_{i_k}}$ indicates the user's rating for resource $r_k \in R$, common in traditional collaborative filtering techniques, but by vectors of interest scores assigned to concepts taken from an ontology $\mathcal{O}$ over resource's concepts. The main idea in representing user profiles in this way is the possibility to exploit the hierarchy of the ontology to generate more overlap and thus allow more meaningful similarity computation. The core idea lies in the users' profile assembly, which is briefly described as follows (see Example 2). For every concept $c \in \mathcal{C}_{\mathcal{O}}$ having the resources r_k that user u_i has implicitly rated as instances, it is also inferred an interest score for all super-concepts of c, where scores assigned to super-concepts decay with increasing distance from the concept c. There are other steps concerning score propagation and normalization (see [27] for more details) that will not be covered here due to space reasons.

Example 2. Consider the ontology depicted in Fig. 1.b. Let u_i have implicitly rated resource r_1. In this case this resource is assigned to just one concept, namely $\{$*emo* $\}$. Now let $s := 100$ denote the overall accorded interest score. After score propagation and normalization, the profile vector for user u_i is composed as $\boldsymbol{u_i} := (\textit{emo} := 53.3, \textit{alternative} := 26.6, \textit{rock} := 13.3, \textit{root} := 6.6)^T$. Note that the interest score (100) is unevenly distributed among the ancestors of emo, where higher concepts receive lower scores, thus reflecting the loss of specificity upwards to the root. After the

Table 1. Characteristics of the knowledge bases. For convenience, we also let $|T|$ represent the count of concepts in musicmoz

dataset	$\|U\|$	$\|T\|$	$\|R\|$	$\|Y\|$
last.fm	3532	7081	982	130899
musicmoz	-	555	982	-

Table 2. Examples of the three best semantic mappings between tags from *last.fm* and concepts from *musicmoz*

electro	*hip hop*	*chillout*	*old skool dance*	*anything else but death*	*depeche mode*
electronica	hip hop	alternative	house	heavy metal	experimental rock
dance	rhythm and blues	rock	hip hop	metal	synthpop
alternative	rap	electronica	electronica	rock	indie

taxonomy-based profiles are assembled for all users, traditional user-based collaborative filtering can be directly applied.

5 Experiments and Discussion

To evaluate our approach we used two different knowledge bases defined over the music domain, where musical resources are assigned to concepts either by the users of a folksonomy or by the experts in the domain. As the folksonomy representative we have chosen *Last.fm*[12], a social tagging system that provides personalized radio stations where users can tag artists and tracks they listen to. Representing the domain-expert we have chosen the *Open Music Project*[13](*musicmoz*), which is based on the *Open Directory*[14] philosophy and aims to be a comprehensive knowledge base about music. We extracted the *style* hierarchy representing a taxonomy of music genres[15] from *musicmoz* to constitute the domain-expert ontology. Since we consider that the aforementioned knowledge bases are defined over the same set of instances, we eliminated all the resources that are not present in both knowledge bases. Table 1 gives a brief overview on the datasets after this pre-processing.

Folksonomy Enrichment. In the semantic mapping step for the folksonomy enrichment described in Section 2, we also handled concept duplications by eliminating tags that are very similar to the concepts that are to be included in the enriched folksonomy. We do this by using the Levenshtein distance metric with a high threshold. Note that this also provides a lexical correction for misspellings, since if we detect that *altertative* and *alternative* are duplicates for example, we just include the syntactically correct one, in

[12] This data can be easily gathered by the Web Services provided by last.fm (http://www.last.fm/api)

[13] http://musicmoz.org

[14] http://www.dmoz.org

[15] The hierarchy is composed by cross references which were disregarded in order to guarantee the tree structure

this case *alternative*. Table 5 illustrates some of the tags and their corresponding three best semantic mappings. While some mappings are trivial, others are very interesting. Consider the tag *old skool dance* for example. The best mapping refers to *house*, which according to *Wikipedia*[16]: "*House music is a style of electronic dance music that was developed by dance club DJs in Chicago in the early to mid-1980s*", thus indeed "old skool dance". This can help both experts and users to evolve and specialize their vocabulary in order to further improve the knowledge sharing. Moreover, looking to the three best mappings of the tag *anything else but death* e. g., it is easy to infer that this tag refers to people who likes all styles of heavy music except the subgenre *death metal*, even though it is not explicit just by looking at the tag. Another interesting example is the tag *depeche mode*. Depeche Mode actually refers to an English band from the 80s, which according to *Wikipedia*[17] belongs to the genres *New Wave*, *Synth Pop*, *Post Punk* and *Alternative Dance*. Note that *synthpop* is the second best map in this case and therefore conforms to other authority sources such as *Wikipedia*.

Collabulary Learning. After some experiments for calibration, we used the following setting for the ontology learning process. The count of iterations was set to 7, the minimum support threshold in the i-th iteration was $\frac{0.025 \cdot |D|}{2^i}$ where $|D|$ is the count of transactions. This led in our case to a minimum support of about 500 in the first iteration, and 8 in the last one. This is reasonable, as we assume that tags occurring more than 500-times together certainly correlate no matter whether they co-occur for example 1100-times or just 700-times. We also assume that in our database of around 40000 transactions, two tags co-occur at least 8-times, if they are correlated. Edge thresholds were chosen to be 1.5 (for itemsets of size 2), 1.4 (for itemsets of size 3), 1.3 (for itemsets of size 4), 1.2 (for itemsets of size 5) and 1.1(for itemsets of size 6). The assumption behind this choice is that at least 50% of the users tagging a resource with a subconcept (subtag), also tags the same resource with the corresponding super-concept. It is also assumed, that a super-concept has at least 3 subconcepts (subtags). This results in the super-concept (supertag) being used at least $3 \cdot 0.5 = 1.5$-times more often than its subconcept (subtag). According to the *Large Itemset Assumption*, relaxed thresholds can be applied for itemsets of size more than 2. To filter "misleading" itemsets, we used a goodness threshold of $g = 2$.

Due to space reasons in Fig. 4 we just show an extract of the learned ontology [18] with root in *emo* and maximum depth set to 2. It is interesting to note that while *emo* is a leaf node in the domain expert ontology, it spans several other subtrees in our case. Looking at the definition of the genre in Wikipedia[19] we see that originally *emo* refers to *punk hardcore* and *indie rock* but starting in the mid-1990s, the term evolved and began to refer to a more melodic and less chaotic kind of *indie rock* style. Notice that as children of *emo* we have both concepts associated to the original definition, which comes from the domain expert-ontology (e. g., *punk hardcore*, *indie rock*) and the ones associated with the term evolution (*nineties* and *post hardcore*). Also notice, that in the

[16] http://en.wikipedia.org/wiki/House_music

[17] http://en.wikipedia.org/wiki/Depeche_Mode

[18] Generated with Pajek (http://pajek.imfm.si/doku.php)

[19] http://en.wikipedia.org/wiki/Emo

definition given in *musicmoz*[20] for the genre, the evolution of the term is not mentioned, denoting that controlled vocabularies evolve somewhat slower than folksonomies, as already intuitively expected.

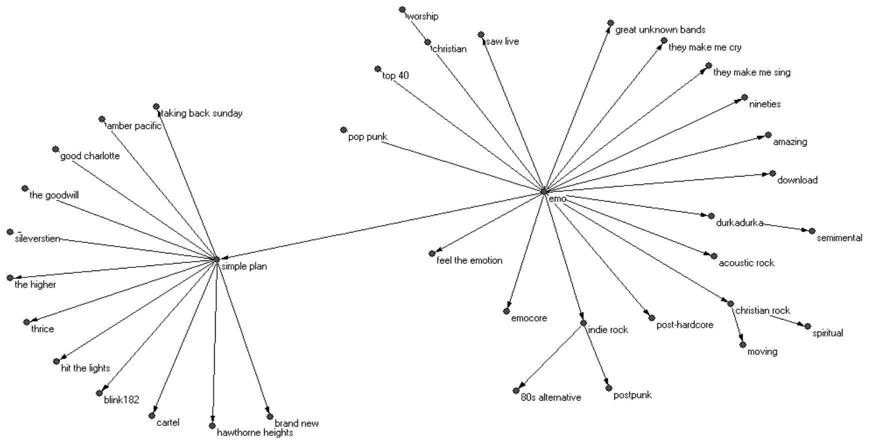


Fig. 4. Extract of the learned ontology

Evaluation. As mentioned in Section 5, we have chosen three ontologies to evaluate according to our benchmark, namely, a trivial one representing the folksonomy (see Section 2) as the baseline, a domain-expert ontology represented by *musicmoz* and the collabulary. We have used a *AllBut1*[8] protocol to evaluate the obtained recommendations where the test set was obtained by randomly selecting one resource from every user. Note that this also means eliminating all the tag assignments of the corresponding user for the respective test resource. The rest of the data is used for the enrichment of the folksonomy and learning the collabulary that will be used by the recommender algorithm. We have repeated this procedure 5 times and averaged the outcomes in order to be more confident about the results[21]. Our evaluations considered any resource in the recommendation set that matches the resources in the testing set as a ”hit”, which is equivalent to the *Recall* metric, typical in such scenarios. The number of recommended resources was set to 10 and the size of the neighborhood to 20.

Fig. 5 shows the results of the experiments. Note that while the domain-expert ontology provides a significant improvement over the baseline, the colabullary largely outperforms both. Intuitively this is explained by the fact that we are not restricting the vocabulary of the user, on the contrary, we provide an enriched ontology composed by the best of both worlds. Based on this empirical evidences we can then conclude that while domain-expert ontologies indeed improve the information finding in comparison to pure flat folksonomies, the learned collabulary helps even further.

[20] http://musicmoz.org/Styles/Rock/Alternative/Emo/

[21] Standard deviation on the top of the bars in Fig. 5

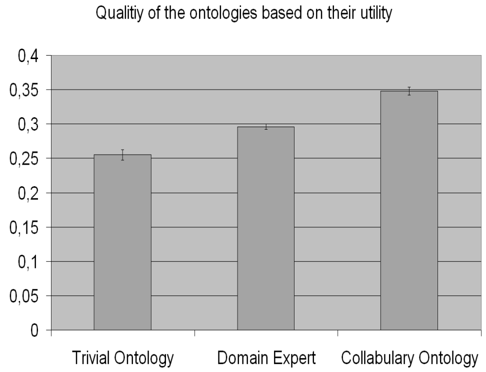


Fig. 5. Ontology evaluation based on recommender systems

6 Related Work

Given the novelty of the problem, there are still very few related works. In [24] e. g., the authors rely on external authority sources or on Semantic Web ontologies to make sense of tag semantics. Even though this can help finding more interesting relations than co-occurrence models, it can somewhat restrict the relation discovery, since if a relation is not defined in these external sources, it is assumed that the tags are not related, even if they frequently co-occur in the dataset. We instead, infer the relations directly from the data and thus are not dependant on external sources.

Other related areas are ontology learning and evaluation. As pointed out before, most of the literature concerning ontology learning from folksonomies base their approaches upon co-occurrence models. Mika [17] e. g., use co-occurrence of tags with resources and users to build graphs relating tags and users and also tags and resources. In [23] conditional probabilities are used to find subsumption relations and [26] use probabilistic unsupervised methods to derive a hierarchy of tags. Given that co-occurrence models are the core subject of frequent itemset mining, in [22] it is proposed the application of association rules between projections of pairs of elements from the triadic context model of folksonomies, although in a conceptual level only. Even though all these works qualitatively contribute for making folksonomies more useful, performance issues are rarely mentioned. Furthermore, all these works are only subjectively evaluated by checking whether the derived ontologies match a reference ontology for example.

7 Conclusions and Future Work

In order to take full advantage of folksonomies' potential to alleviate the knowledge bottleneck that slows down the Semantic Web realization, one needs to educate social tagging systems' users towards clear annotation, without however, taking out their freedom to tag. In this paper we proposed an approach to address this issue by combining the user vocabulary with the expert vocabulary in an integrated fashion. Various studies have shown (e. g., [10]) that the vocabulary of users in social tagging systems stabilize over time due to exposure to each other tags and resources. Taking this into account, we argue that exposing users to a collabulary where meaningful concepts are matched and put together with tags, have the potential to make the whole vocabulary converge to a more meaningful, shareable and useful knowledge representation. Furthermore, we have empirically evaluated the extent to which folksonomies, domain-expert ontologies and collabularies help recommender systems to deliver useful information to the users. Looking at the results, the main lessons learned were: (i) indeed hierarchies provide advantages over flat folksonomies, and (ii) collabularies provide clear benefits over pure domain-expert ontologies.

The main contributions can be summarized as follows: 1. The proposal of a new approach to address the trade-off between folksonomies and domain-expert ontologies. 2. The proposal of a new algorithm for ontology learning from folksonomies based on frequent itemset mining techniques. 3. The proposal of a new benchmark for ontology evaluation. 4. The evaluation of the proposed model on real-life data, namely, from *Last.fm* and *musicmoz*. In future work we plan to extend the method here proposed with the identification of non-taxonomical relations between the tags of the enriched folksonomy.

Acknowledgement. This work is supported by CNPq, an institution of Brazilian Government for scientific and technologic development, and the X-Media[22] (IST-FP6-026978) project.

References

1. Wikipedia article. http://en.wikipedia.org/wiki/Folksonomy. Accessed on May 2008.
2. R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *Proc. of SIGMOD 1993*, pages 207–216. ACM Press, May 1993.
3. R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Proc. of the 20th international conference on Very Large Data Bases (VLDB'94)*, pages 478–499. Morgan Kaufmann, September 1994.
4. T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, May 2001.
5. F. Bodon. A fast apriori implementation. In *Proc. 1st IEEE ICDM Workshop on Frequent Item Set Mining Implementations*, volume 90 of *CEUR Workshop Proc.* CEUR-WS.org, 2003.
6. C. Borgelt. Efficient implementations of apriori and eclat. In *FIMI*, volume 90 of *CEUR Workshop Proc.* CEUR-WS.org, 2003.
7. C. Borgelt. Recursion pruning for the apriori algorithm. In *FIMI*, volume 126 of *CEUR Workshop Proc.* CEUR-WS.org, 2004.
8. J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI-98)*, pages 43–52. Morgan Kaufmann, 1998.
9. C. H. Brooks and N. Montanez. Improved annotation of the blogosphere via autotagging and hierarchical clustering. In *WWW '06: Proc. of the 15th international conference on World Wide Web*, pages 625–632, New York, NY, USA, 2006. ACM.
10. C. Cattuto, V. Loreto, and L. Pietronero. Collaborative tagging and semiotic dynamics, May 2006. http://arxiv.org/abs/cs/0605015.

[22] http://www.x-media-project.org,

11. H. Chalupksy. Ontomorph: A translation system for symbolic knowledge. In *Proc. of the 17th International Conference on Knowledge Representation and Reasoning*, 2000.
12. P. Cimiano, A. Hotho, and S. Staab. Learning concept hierarchies from text corpora using formal concept analysis. *Journal of Artificial Intelligence Research (JAIR)*, 24:305–339, 2005.
13. A. Doan, J. Madhavan, P. Domingos, and A. Y. Halevy. Ontology matching: A machine learning approach. In *Handbook on Ontologies*, International Handbooks on Information Systems, pages 385–404. Springer, 2004.
14. A. Goldenberg and A. Moore. Tractable learning of large bayes net structures from sparse data. In *Proc. of the 21st International Conference on Machine Learning*, 2004.
15. P. Heymann and H. Garcia-Molina. Collaborative creation of communal hierarchical taxonomies in social tagging systems. Technical Report 2006-10, Stanford University, April 2006.
16. A. Hotho, R. Jaeschke, C. Schmitz, and G. Stumme. Information retrieval in folksonomies: Search and ranking. In *The Semantic Web: Research and Applications*, volume 4011 of *Lecture Notes in Computer Science*, pages 411–426, Heidelberg, June 2006. Springer.
17. P. Mika. Ontologies are us: A unified model of social networks and semantics. In *The Semantic Web - ISWC 2005, Proceedings of the 4th International Semantic Web Conference, ISWC 2005, Galway, Ireland, November 6-10*, volume 3729 of *Lecture Notes in Computer Science*, pages 522–536. Springer, 2005.
18. N. F. Noy and M. A. Musen. Prompt: Algorithm and tool for automated ontology merging and alignment. In *AAAI/IAAI*, pages 450–455, 2000.
19. J. Pei, J. Liu, and K. Wang. Discovering frequent closed partial orders from strings. *IEEE Transactions on Knowledge and Data Engineering*, 18(11):1467–1481, 2006.
20. R. Porzel and R. Malaka. A task-based approach for ontology evaluation. In *Proc. of ECAI 2004 Workshop on Ontology Learning and Population*, Valencia, Spain, August 2004.
21. P. Resnick, N. Iacovou, M. Suchak, P. Bergstorm, and J. Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proc. of ACM 1994 Conference on Computer Supported Cooperative Work*, pages 175–186, Chapel Hill, North Carolina, 1994. ACM.
22. C. Schmitz, A. Hotho, R. Jaeschke, and G. Stumme. Mining association rules in folksonomies. In *Data Science and Classification: Proc. of the 10th IFCS Conf.*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 261–270, Berlin, Heidelberg, 2006. Springer.
23. P. Schmitz. Inducing ontology from flickr tags. In *Proc. of the Workshop on Collaborative Tagging at WWW2006*, Edinburgh, Scotland, May 2006.
24. L. Specia and E. Motta. Integrating folksonomies with the semantic web. In *Proc. of the European Semantic Web Conference (ESWC2007)*, volume 4519 of *LNCS*, pages 624–639, Berlin Heidelberg, Germany, July 2007. Springer-Verlag.
25. K. Sriphaew and T. Theeramunkong. A new method for finding generalized frequent itemsets in generalized association rule mining. In *ISCC '02: Proc. of the Seventh International Symposium on Computers and Communications (ISCC'02)*, page 1040, 2002.
26. M. Zhou, S. Bao, X. Wu, and Y. Yu. An unsupervised model for exploring hierarchical semantics from social annotations. In *Proc. of the 6th International Semantic Web Conference and 2nd Asian Semantic Web Conference (ISWC/ASWC2007), Busan, South Korea*, volume 4825 of *LNCS*, pages 673–686. Springer Verlag, November 2007.
27. C. Ziegler, L. Schmidt-Thieme, and G. Lausen. Exploiting semantic product descriptions for recommender systems. In *Proc. of the 2nd ACM SIGIR Semantic Web and Information Retrieval Workshop (SWIR '04)*, Sheffield, UK, 2004.