A Hybrid Approach to Constructing Tag Hierarchies

Geir Solskinnsbakk and Jon Atle Gulla

Department of Computer and Information Science,
Norwegian University of Science and Technology,
Trondheim, Norway
{geirsols,jag}@idi.ntnu.no

Abstract. Folksonomies are becoming increasingly popular. They contain large amounts of data which can be mined and utilized for many tasks like visualization, browsing, information retrieval etc. An inherent problem of folksonomies is the lack of structure. In this paper we present an unsupervised approach for generating such structure based on a combination of association rule mining and the underlying tagged material. Using the underlying tagged material we generate a semantic representation of each tag. The semantic representation of the tags is an integral component of the structure generated. The experiment presented in this paper shows promising results with tag structures that correspond well with human judgment.

1 Introduction

The collaborative effort of users tagging resources is often referred to as a *folksonomy* [1]. Generally one can say that a folksonomy consists of three entities; the *user*, the *tag*, and the *resource* [1]. Tags used in the action of tagging a resource are not necessarily bound to dictionaries or thesauri and can be created by the users themselves. As for the resources, generally anything with a URL can be tagged, web pages, images, presentations, etc. Examples of popular sites that employ tagging are among others *Delicious*[1] and *Flickr*[2]. Delicious lets users tag resources on the web, while Flickr lets users tag their own images. When a user tags a resource with a set of tags in Delicious, the combined information is called a bookmark.

Folksonomy tags do not generally have any associated structure. For visualization purposes however, tag clouds or lists of related tags can be employed and presented to users. The intention of this paper is to generate a semantic structure of tags in the folksonomy that can later support semantic information access. The idea is to combine tag structure and traditional ontologies in a unified semantic search framework.

[1] http://delicious.com/
[2] http://www.flickr.com/

R. Meersman et al. (Eds.): OTM 2010, Part II, LNCS 6427, pp. 975–982, 2010.

Most of the research on structuring folksonomies is based on the *tags*, *users*, and *tagged resources* alone. In our opinion, the textual content of the tagged resources themselves should also be taken into account and can give important input to the process of structuring the folksonomy. In this paper we use a combination of the *tags* and *tagged resources* to structure the folksonomy. In our research we have used Delicious as the sample data, while the general approach should be viable for any folksonomy which is based on tagging textual resources.

Our approach to generating a semantic structure of the folksonomy data builds a hierarchical structure over the tags and associates every tag in the folksonomy with a vector of semantically related terms. The structure is mainly guided by association rule mining of the tags in the bookmarks. We have then explored if there are any defining properties among the tag vectors that can help us verify the type of relation between the tags in the hierarchy. The results of our experiment seem promising with a high ratio of tag relations that are classified by the test subjects as either related or correctly related.

2 Related Work

In [2], Mika presents an approach for generating lightweight ontologies based on processing a tripartite hypergraph representation of the folksonomy. Heymann et al. [3] describe an algorithm for generating structure based on representing the tags as vectors and calculating the cosine similarities among the vectors. During hierarchy construction the tags are added to the hierarchy by considering the cosine similarity and tags are processed in order of centrality in a similarity graph. Benz et al. [4] presents an approach (extension of [3]) in which each concept is represented by one or more folksonomy tags. Zhou et al. [5] present an approach based on clustering, finding relations between clusters of tags. Specia et al. [6] describe an approach based on clustering and using external information sources (WordNet [7] and Google[3]) to assist in the structuring of the tags. Laniado et al. [8] present an approach that relies on WordNet for disambiguation and is used as a basis for structuring the tags. Schwarzkopf et al. [9] performs some experiments based on previous structuring approaches and modifies them by imposing additional requirements on similarity to filter some of the relations. In [10], Schmitz et al. derive two-dimensional views of the folksonomy data and apply association rule mining to the reduced data. Lin et al.. [11] mine association rules between pairs of tags, and structure is imposed using WordNet as a guide. Further rules are filtered based on the cosine similarity, in which the tags are represented as vectors of resources with binary weight. Our approach is similarly based on association rules for structure and cosine similarity for validation. The tag vector representation in our research is more detailed by taking into account the textual content of the documents. Further, we do not use any external knowledge sources like WordNet, since domain-specific information access normally requires vocabularies to specific for WordNet to be appropriate.

[3] `http://www.google.com`

3 Tag Vectors

We have previously done some work on extending ontologies with vectors giving a semantic representation of the concepts of the ontology and using them for information retrieval (see [12]). In line with our previous work, we are now looking at constructing similar vectors for tags in a folksonomy.

The folksonomy is a collection of triples [13] (*URL*, *tag*, *user*) where the interpretation is that the user, u_i, has tagged a resource (URL), r_j, with the tag t_k. The fact that a user has opted to tag a given resource with a specific tag means that in the user's view, the tag is a representative term/word to describe the resource. Each tag vector constructed by our approach can be interpreted as a semantic representation of the tag. The weight assigned to each term in the tag vector reflects two aspects; (1) the importance of the term with respect to the tag (internal representation) and (2) the ability the tag has to discriminate this tag from other tags (external representation). To achieve this we use the $tf \cdot idf$ measure [14]. The terms in the tag vector together with the term weights give a representation of the tag that reflects how the tag is applied by the users of the bookmarking service. Our main motivation behind these vectors is twofold; (1) we want to use them as a step in the construction of the structure to assure higher quality of the structure, and (2) we want to use the vectors for information access in a method that is compatible with the use of ontological profiles (see [12]). The definition of a tag vector is given as Definition 1.

Definition 1. *Tag Vector. Let V be the set of n terms (vocabulary) in the collection of tagged resources. $t_i \in V$ denotes term i in the set of terms. Then the tag vector for tag j is defined as the vector $T_j = [w_1, w_2, \ldots, w_n]$ where each w_i denotes the semantic relatedness weight for each term t_i with respect to tag T_j.*

4 Approach

Our approach assumes that we are dealing with a folksonomy that has been built by users tagging textual resources. The process is based on three phases; (1) *association rule mining*, (2) *hierarchy construction*, and (3) *structure confirmation* using the tag vectors.

4.1 Association Rule Mining

The first step in the process is to run association rule mining on the set of tags, using the Apriori algorithm by Agrawal and Srikant [15]. The Apriori algorithm uses prior knowledge to reduce the search space when mining for association rules. This is done by first generating the set of frequent 1-item sets (i.e. sets containing a single item). The frequent 1-item sets are used as input (prior knowledge) for generating the frequent 2-item sets, since every frequent 2-item sets must be a combination of elements from the frequent 1-item sets [16].

In our approach, we are only interested in association rules that are generated from the frequent 2-item sets, so the algorithm is terminated at this point. The

reason for this is that the interpretation of rules from the frequent 2-item sets is easier than for n-item sets (for n>2).

Association rules take on the form $T_i \rightarrow T_j$, where T_i is the premise, and T_j is the consequence. This rule states that if T_i is observed, one can with a certain probability observe T_j. Within the context of bookmarking and folksonomies, this can be translated into the interpretation: Whenever tag T_i is observed, one can with a certain probability observe the tag T_j in the same bookmark. Measures that are highly important for association rules (and in fact used to filter the rules) are support and confidence. Support is the number of times an observation occurs in the data set, while confidence is the percentage of observations that contain the premise and that also contain the consequence (see Equation 1 [16]).

$$confidence(T_i \rightarrow T_j) = P(T_j|T_i) = \frac{supportCount(T_i \cup T_j)}{supportCount(T_i)} \tag{1}$$

The data used as basis for the association rules is the bookmarking data from the folksonomy. Each set of tags used as basis in the association rule mining process consists of all tags assigned to a single resource by a single user, i.e. a single bookmark. We are employing minimum support and minimum confidence measures for the mined rules (see Section 5).

4.2 Hierarchy Construction

The first phase of our approach results in a set of association rules. Our initial interpretation is that the premise of the rule may be viewed as a child of the consequence. The motivation behind this interpretation is best described with an example. Assume two tags, A and B, and a set of bookmarks tagged with B. If a subset of the resources tagged with B also contain the full set of bookmarks tagged with A, we regard A as a specialization of B, and the rule with the highest confidence is $A \rightarrow B$. In other words, A is a subclass of B.

The construction of the hierarchy starts with an empty root node. Next we find all consequences which do not appear as premises of any rule. These are added as direct children of the root node. For each first level child, all tags that appear as premises of rules with the first level child as consequence are added as children. This process continues until there are no more children to add, or an attempt is made to add a tag that already exists in the path from the root to the current node.

4.3 Structure Verification

We now need to generate tag vectors of all the tags in the system. The first step is to collect the relevant textual content for each tag. The text relevant to a specific tag is taken to be the sum of all documents that have been allocated this tag. The first part of this phase is constructing a tag vector, which can be interpreted as a semantic description of the tag. This is done by using the textual

content of the tagged resources. The textual content is then preprocessed, where we remove any markup like html, if present, remove stop words, and stem the terms using the porter stemming algorithm [17].

The next step is to generate the basic tag vectors. This is done for each tag, adding all terms remaining after preprocessing that occur in resources tagged by the tag in question. We also employ some weighting function which reflects the frequency of applying the tag to each resource. This means that if a tag has been used to tag one resource 10 times and another 3 times, the first should be more important to the tag vector than the second. We argue that the use of the tag on the first resource is in greater agreement among the users, which should be reflected in the tag vector. It might however be the case that the resource tagged 3 times has been tagged a much higher number of times in total (with other tags) than the resource that has been tagged 10 times. In light of this it could seem reasonable to add this to the equation, but in our view, we are trying to represent how the public have used each tag, and thus the tagging frequency of a resource with other tags is not that interesting. The basic weight of the terms in the tag vector at this stage is calculated according to Equation 2, where $tv_{i,j}$ is the weight of term i in the tag vector of tag j, $\alpha_{j,r}$ is the number of times tag j has been used to bookmark resource r, $f_{i,r}$ is the frequency of term i in resource r, and R_j is the set of resources tagged with tag j.

$$tv_{i,j} = \sum_{r \in R_j} \alpha_{j,r} \cdot f_{i,r} \tag{2}$$

This basic vector is a good description of the internal representation of the tag, while we also need the vector to discriminate against other tags. This is done by calculating the final weight of the terms in the vectors according to the $tf \cdot idf$ score. The calculation is shown as Equation 3, where $tfidf_{i,j}$ is the tfidf score for term i in the vector for tag j, $tv_{i,j}$ is the result of Equation 2, N is the number of tag vectors, and n_i is the number of tag vectors containing term i. Lastly the tag vectors are normalized to unit length.

$$tfidf_{i,j} = \frac{tv_{i,j}}{max(tv_{l,j})} \cdot \log \frac{N}{n_i} \tag{3}$$

The similarity between two tag vectors is found using standard cosine similarity calculations [14]. Unlike association rules, the cosine similarity does not give any direction of the relation, only a numeric value of the strength of the relation. Thus, the cosine similarity is used as a supporting tool for confirming the structures that have been built from the association rules.

4.4 Interpretation of Confidence and Cosine Similarity

Confidence. *As the confidence of a rule used as basis for a relation increases, so does the quality of the relation.* The motivation for this interpretation is that a higher confidence in theory means that the probability for the connection is

higher; thus we expect to see that a high confidence will lead to a high probability for a good or correct relation.

Cosine Similarity. *The cosine similarity between two tags can give us supporting information on how two tags are related.* The motivation for this interpretation is to see whether the cosine similarity can give any information in addition to the information from the association rule that will help us interpret the relation. E.g. a high cosine similarity could point in the direction of synonyms, and a low value in the direction of no relation.

5 Preliminary Results

Our initial experiment is based on a data set from Delicious which we crawled between December 2009 and January 2010. The data set consists of 195471 bookmarks tagging resources in the English section of Wikipedia ("http://en.wikipedia.org/wiki") which were mapped to a cleaned and Part-of-Speech tagged Wikipedia dump from June 2008 [18]. This dump has been used as the textual foundation for constructing the tag vectors.

In the experiment we set a confidence threshold of 0.4, and a support threshold of 50, which left us with 1752 tags resulting in 771 association rules. The structures we chose for evaluation were chosen based on requiring the depth of the tree to be at least 3 (not counting the root). This left 40 trees and 303 relations to evaluate (for an example see Figure 1). The evaluation consisted of having 9 colleagues evaluate the relations. The relations could be described as *correct* (correct hierarchy relation), *related* (correct relation, but not hierarchical or reverse hierarchical), *equivalent* (synonym), *not related*, and *unknown* (the evaluator does not recognize the meaning of the tag(s)). On average, 31.7% of the relations in the hierarchy received the classification *related*, while 43.7% received the classification *correct*. Only 5.3 % of the relations were classified as *not related*, 13.2% as *unknown*, and 6.1% as *equivalent*. This points out to us that the approach gives a generally good quality hierarchy. One aspect that needs further attention is that relatively many (31.7%) of the relations were classified as related (correct non-taxonomic or reverse taxonomic relations). This may be due to the test subjects having different views on the relations in terms of how the relations should be modeled.

From the evaluation we also found that the cosine similarity between the tags were generally helpful in locating relations classified by the test subjects as *equivalent*, with an average score ranging from 0.78 to 0.86 (depending on parameter settings) with a stable difference of approximately 0.2 to the next highest scoring relation type. The average value of the cosine score between tags of the other relation types did not give any indication of relation type. In fact, *correct* and *related* tag relations had approximately equal cosine similarities, while *non related* relations had marginally lower scores (difference in the range 0.02-0.05).

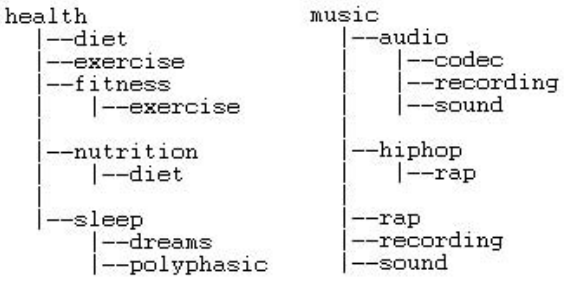

Fig. 1. Sample hierarchies from the experiment rooted at health and music (evaluated versions)

Looking at the confidence scores we found that higher confidence levels seem to produce a higher ratio of *correct* relations, a lower ration of *related* relations, and a small increase in the ratio of *non related* tag relations.

Overall the approach seems promising, although more work would be needed to improve the overall quality of the approach.

6 Conclusion

In this paper we have described an approach for creating a semantic structure based on folksonomies. The main structure is based on association rule mining of the tag set. We have also introduced the concept of tag vectors as semantic representations of the tags in the folksonomy, and how they can be used to evaluate the quality of the structure generated.

Our experiment based on a data set from delicious containing 195471 bookmarks, and the evaluation of the structure based on a minimum support count of 50, shows promising results. Looking at the combined results for the relations classified by the test subjects as *related* and *correct*, the results show that on average 75.4% of the evaluated relations were by the test subjects described as such. Only 5.3% of the relations were classified as *not related*. This seems to point in the direction that our approach based on co-tags is good for generating ontological structures based on folksonomies. The cosine similarity between tag vectors also seem to be a good tool to identify equivalent tags. We will continue to improve the approach to try to find ways of isolating and removing relations that are not correct. We will also in our future work use the semantic tag structure for information access.

Acknowledgment. This research was carried out as part of the IS_A project, project no. 176755, funded by the Norwegian Research Council under the VERDIKT program.

References

1. Vander Wal, T.: Folksonomy coinage and definition, http://vanderwal.net/folksonomy.html (accessed March 2010)
2. Mika, P.: Ontologies are us: A unified model of social networks and semantics. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 522–536. Springer, Heidelberg (2005)
3. Heymann, P., Garcia-Molina, H.: Collaborative Creation of Communal Hierarchical Taxonomies in Social Tagging Systems, InfoLab Technical Report, Stanford (2006)
4. Benz, D., Hotho, A., Stützer, S., Stumme, G.: Semantics made by you and me: Self-emerging ontologies can capture the diversity of shared knowledge. In: Proceedings of the 2nd Web Science Conference, Raleigh, NC, USA (2010)
5. Zhou, T.C., King, I.: Automobile, Car, and BMW: Horizontal and Hierarchical Approach in Social Tagging Systems. In: Conference on Information and Knowledge Management, Proceeding of the 2nd ACM Workshop on Social Web Search and Mining, Hong Kong, China (2009)
6. Specia, L., Motta, E.: Integrating Folksonomies with the Semantic Web. In: Franconi, E., et al. (eds.) ESWC 2007. LNCS (LNAI), vol. 4519, pp. 624–639. Springer, Heidelberg (2007)
7. Fellbaum, C. (ed.): WordNet: An Electronic Lexical Database. MIT Press, Cambridge (1998)
8. Laniado, D., Eynard, D., Colombetti, M.: Using WordNet to turn a folksonomy into a hierarchy of concepts. In: Proceedings of SWAP 2007, the 4th Italian Semantic Web Workshop, CEUR Workshop Proceedings, Bari, Italy, December 18-20 (2007), http://ceur-ws.org/Vol-314/51.pdf
9. Schwarzkopf, E., Heckmann, D., Dengler, D., Kröner, A.: Mining the Structure of Tag Spaces for User Modeling. In: Workshop on Data Mining for User Modeling (International Conference on User Modeling 2007) (2007)
10. Schmitz, C., Hotho, A., Jäschke, R., Stumme, G.: Mining Association Rules in Folksonomies, Data Science and Classification. In: Proc. of the 10th IFCS Conf., Studies in Classification, Data Analysis, and Knowledge Organization (2006)
11. Lin, H., Davis, J., Zhou, Y.: An Integrated Apporoach to Extracting Ontological Structures from Folksonomies. In: Arayo, L., et al. (eds.) ESWC 2009. LNCS, vol. 5554, pp. 654–668. Springer, Heidelberg (2009)
12. Solskinnsbakk, G., Gulla, J.A.: Ontological Profiles in Enterprise Search. In: Gangemi, A., Euzenat, J. (eds.) EKAW 2008. LNCS (LNAI), vol. 5268, pp. 302–317. Springer, Heidelberg (2008)
13. Heymann, P., Koutrika, G., Garcia-Molina, H.: Can Social Bookmarking Improve Web Search? In: First ACM International Conference on Web Search and Data Mining (WSDM 2008), Stanford, CA, February 11-12 (2008)
14. Baeza-Yates, R., Ribeiro-Neto, B.: Modern Information Retrieval. ACM Press, New York (1999)
15. Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules. In: Proceedings of the 20th International Conference on Very Large Databases (September 1994)
16. Han, J., Kamber, M.: Data Mining: Concepts and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2006)
17. Porter, M.F.: An algorithm for suffix stripping. Program 14(3), 130–137 (1980)
18. Artiles, J., Sekine, S.: Tagged and Cleaned Wikipedia (TC Wikipedia), http://nlp.cs.nyu.edu/wikipedia-data/ (accessed December 2009)