On Deriving Tagsonomies: Keyword Relations Coming from Crowd

Michal Barla and Mária Bieliková

Institute of Informatics and Software Engineering, Faculty of Informatics
and Information Technologies, Slovak University of Technology
Ilkovičova 3, 842 16 Bratislava, Slovakia
{Name.Surname}@fiit.stuba.sk

Abstract. Many keyword-based approaches to text classification, information retrieval or even user modeling for adaptive web-based system could benefit from knowledge on relations between various keywords, which gives further possibilities to compare them, evaluate their distance etc. This paper proposes an approach how to determine keyword relations (mainly a parent-child relationship) by leveraging collective wisdom of the masses, present in data of collaborative (social) tagging systems on the Web. The feasibility of our approach is demonstrated on the data coming from the social bookmarking systems delicious and CiteULike.

1 Introduction

Phenomenon of the Social Web, with its roots in Web 2.0, is gaining a lot of attention all over the world in both research and practice. We are studying the power and wisdom of masses, when millions of people switched from the passive reading of the content to the active participation in its creation. People are blogging, sharing wikis, connecting themselves in various social applications and above all – tagging almost everything they get into touch: bookmarks, photographs, videos, publications, blogposts, articles etc. People got used to classify items by assigning few simple tags to it and to use those tags for a future retrieval of their favorite items. More, they are often expecting to find a new, yet unseen content by using their own tags. A part of the Web 2.0 success lies in an implicit agreement of masses on a shared (but never explicitly defined) vocabulary used to tag items – folksonomies.

At the same time, vast amount of content requires efficient navigation support, content reorganization or filtering – personalization and adaptation of the web. Its efficiency is dependent on adaptive system's ability to capture and maintain user model. A lot of research was devoted to finding the most suitable, flexible or most generic and all-encompassing user model representation [1, 2], however, so far we are not aware of any explicit agreement on an ideal model representation.

The obvious question, when analyzing the success of Web 2.0, is whether an assignments of keywords (tags) to *user* instead of to *pages* (i.e., creation of tag-based user model) could lead to simple, viable and efficient approach to user modeling for adaptive web-based system. The challenge is then to combine the user model with the models of communities coming from the emerging social web and create a solid platform for personalization based on both traditional (e.g., as presented in [3]), and social approaches, such as the one presented in [4].

When considering tag-based user models in a (tag-based) Web 2.0 environment, we are facing the need to be able to compare various tags. We might need to compare user characteristics or even whole user models, represented by tags, to find similarities between users, which could serve for model maintenance as well as for more complex tasks such as online community creation or recommending in a recommender system. We might also need to compare the domain items represented by tags (e.g., a web page) in order to evaluate the explicit or implicit feedback [5] and update the user model appropriately.

In this paper, we propose a method for inferring various relationships between tags, which allows for full-blown usage of the tag-based user model for personalization and adaptation on the Web.

The paper is structured as follows: In section 2 we explain our approach to finding relationships between tags. Section 3 presents data we acquired and results of experiments we performed. In section 4 we summarize the related works, which served as an inspiration for our algorithm for building hierarchies from folksonomies. Finally, we give conclusions.

2 Finding Relationships between Tags

Our approach to finding relationships between tags combines three distinct approaches:

1. Deriving of parent-child relationships between tags from a given folksonomy;
2. Determining similarity between tags by applying spreading activation on the top of the folksonomy graph;
3. Interconnecting tags by additional semantic relationships as well as enriching the whole tag corpus by adding external keywords; both coming from the Wordnet lexical database.

2.1 Building Hierarchies from Folksonomies

Folksonomy is defined as a hypergraph [6] $H := \langle V,\ E \rangle$, where the set of vertices $V = A \cup T \cup I$ and $A \cap T = \emptyset,\ A \cap I = \emptyset,\ T \cap I = \emptyset$ and the set of ternary edges $E = \{(a, t, i) \mid a \in A,\ t \in T,\ i \in I\}$. A social tagging system can be represented by such a hypergraph with following definitions of the sets A, T and I (we will use them for the rest of the paper as well):

- Actors (users) $A = \{a_1, ..., a_k\}$
- Tags (keywords, concepts) $T = \{t_1, ..., t_l\}$
- Items (objects, instances) $I = \{i_1, ..., i_m\}$

The aforementioned hypergraph can be reduced into three bipartite graphs: a graph holding the associations between actors and tags (AT), actors and items (AI) and tags and items (TI). For example, the AT valued bipartite graph is defined as follows: $AT := \langle A \times T,\ E_{at} \rangle$, $E_{at} = \{(a,\ t) \mid \exists i \in I : (a,\ t,\ i) \in E\}$.

Each such bipartite graph $XY := \langle X \times Y,\ E_{xy} \rangle$ can be furthermore folded into two one-mode graphs $G_X := \langle V_x,\ E_x \rangle$, $G_Y := \langle V_y,\ E_y \rangle$, where $V_x = X$, $V_y = Y$, and $E_x = \{(a,\ b) \mid a,\ b \in X,\ \exists y \in Y : (a,\ y) \in E_{xy},\ (b,\ y) \in E_{xy}\}$, $E_y = \{(a,\ b) \mid a,\ b \in Y,\ \exists x \in X : (x,\ a) \in E_{xy},\ (x,\ b) \in E_{xy}\}$. We can furthermore define a weight of an edge $e_{ab} \in E_i$ in such a one-mode graph as $w(e_{ab}) := |\{k \mid k \in K : (a,k) \in E_{ik},\ (b,k) \in E_{ik}\}|$. In other words, the weight $w(e_{a,b})$ shows the number of times the a and b were linked together in an original bipartite graph.

By folding aforementioned bipartite graphs, we get six different one-mode graphs, each representing different semantic network encoded in the folksonomy. For example, by folding AT graph through tags, we get a social network of actors (users) based on overlapping sets of tags, where the links are between people who have used the same tags with weights showing the number of tags they have used in common. Similarly, we can get a social network based on overlapping sets of items (two people are linked if they have tagged the same item, with weight showing the number of items they have tagged in common). In our work, we were primarily interested in folding TI graph through items, giving us a semantic network of tags.

The proposed approach to creation of hierarchy from the folded folksonomy hypergraph as defined above is based on a rather simple assumptions of set theory:

> *In an ideal situation, the tag t_a is a parent of tag t_b if the set of entities (persons or items) classified under t_b is a subset of the entities under t_a.*

In other words, all items classified under narrow tag also appear under the broader tag.

Moreover, since our goal was to produce a reusable hierarchy of tags, which could be mapped to users' interests and use this hierarchy as a basis for reasoning on those interests, we were not interested in tags, which were used only by a small amount of users, even if they were using it quite extensively. We wanted only what "crowd agrees upon" and were filtering-out tags not achieving a certain degree of popularity (i.e., it is not used by at least k% of all users), even if this decision reduced drastically the amount of tags in the resulting hierarchy.

The algorithm 1 shows the basic idea of our approach using a simple pseudocode. First, we create an artificial *root* of the hierarchy and put it in the set of already processed tags (*ordered tags* in the algorithm). Then, we process all tags from the folksonomy in the following manner:

1. If the tag t does not reach the popularity threshold, we omit it immediately and pass on to the next one.
2. Otherwise, we compute its overlap (intersect) with every tag from the *ordered tags* set, resulting in identifying the tag t_o with a maximum overlap.

3. The parent-child or sibling relationship is established if the ratio of maximum overlap to the overall use of the tag t reaches the pre-defined threshold. The roles of the tags in relationship is determined as follows: if the tag t_o is used significantly more often than t, we declare t to be a child of t_o and vice versa. If the usage of both tags is more or less equal, we declare them as siblings.
4. Before the assignment of relationships, we check whether it will not "break" the context in the hierarchy, i.e., we compute an average overlap of all ancestors or children in the branch respectively. If the average overlap falls below the threshold, we create a duplicate of tag t_o, assign tag t to it appropriately and make a new branch of it. The creation of the duplicate aims at solving the homonymy problem, where the t_o tag has multiple meanings, depending on a context.

2.2 Finding Related Tags by Spreading Activation

Spreading activation is a method for associative retrieval [7] in associative and semantic networks, hence a network data structure consisting of nodes and links modeling relationships between nodes. Searching is done by activating the selected node with an activation energy and spreading this energy through the edges to its *neighbors* so that

$$Energy[neighbor_i] = \frac{Energy[origin]}{|neighbors|}$$

The process runs recursively until convergence. At the end, the nodes' activation levels represent the measure of similarity to the initially chosen node or nodes.

Spreading activation search in the folksonomy finds new relationships, which are not "visible" when considering only set theory assumptions and the algorithm 1. If these relations are added to the already existing parent-child relationships, it allows us to make contextual "jumps" between the tags, which we believe could have an interesting impact for the tag-based user modeling and personalization.

Since the folksonomy does not provide direct links between tags, the spreading activation is performed on either the bipartite graph TI (tags-items) or TA (tags-actors) or on a combination of the two, all depending on the definition what *neighbors* means for the algorithm (i.e., the spreading activation on the TA graph is performed if the *neighbors* of a tag are all actors which used that tag). However, due to the vast amount of connections present in every social tagging system, which leads to enormous time and space complexity of the recursive process, we needed to select a sub-graph of the whole folksonomy. For instance, for the TI bipartite graph, we select the sub-graph by modifying the *neighbors* function of a tag/item so that it will return only popular items/tags, where popularity of a tag or item is defined (following the same principles as in algorithm 1) as a ratio of actors which used that particular tag or tagged that particular item. When spreading through TA graph, we define an actor as popular if he or she used at least $k\%$ of popular tags and tagged at least $l\%$ of popular items.

Algorithm 1: Creation of tag hierarchy

```
Data: a folksonomy
Result: Popular tags structured in a hierarchy
create $root$
$ordered\ tags \leftarrow root$
for each tag $t$ do
    $t.popularity = \frac{|t.users|}{|overallusers|}$
    if $t.popularity < popularity\ threshold$ then
        continue
    end
    $t_o \leftarrow max(\ t \cap tag_x, \forall tag_x \in ordered\ tags\ )$
    if $\frac{|t_o|}{|t|} > overlap\ threshold$ then
        if $|t_o| \gg |t|$ then
            if $rightcontext?(t_o.ancestors, t)$ then
                $t_o.children \leftarrow t$
            end
            else
                create a copy $t_{oc}$ of $t_o$
                $root.children \leftarrow t_{oc}$
                $t_o.children \leftarrow t$
            end
        end
        else if $|t_o| \ll |t|$ then
            if $rightcontext?(t_o.children, t)$ then
                $t.parent = t_o.parent$
                $t.children \leftarrow t_o$
            end
            else
                create a copy $t_{oc}$ of $t_o$
                $t.children \leftarrow t_{oc}$
                $root.children \leftarrow t$
            end
        end
        else
            $t_o.sibling \leftarrow t$
        end
    end
    else
        $root.children \leftarrow t$
    end
end
```

2.3 Applying Knowledge on Keywords from Wordnet

Wordnet [8] (`wordnet.princeton.edu`) is a large lexical database for the English language developed and maintained by Cognitive Science Laboratory at Princeton University. It groups nouns, verbs, adjectives and adverbs into sets of cognitive synonyms (synsets). Moreover, it provides conceptual-semantic and lexical relations between synsets, i.e., one can browse for hyponyms and hypernyms[1] of a given word, other similar words or even antonyms.

The reason why we did not use Wordnet as our source of keywords and relationships at the first place comes from our plans to leverage our hierarchy for user modeling purposes. Tags acquired from social tagging systems are closer to the user than Wordnet, they are more "webbish". Even more, some relationships which emerge from the social tagging systems could *never* come out of Wordnet (for example, a relationship between words `ie`, `png` and `bugs`, pointing to the well-known problem of Internet Explorer's broken PNG support). Relationships acquired from social tagging systems are like ordinary people used them, not like linguistics have decided them to be.

However, we believe, that Wordnet can still significantly contribute to the quality of the tag-based models built upon the generated tag hierarchy. Information on semantic relationships between words can be used not only to identify particular subtrees, which should be merged (in case of synonymy) or divided (because of ambiguity), but also to add new words (along with their relationships) to the hierarchy, which would raise the probability that we will be able to map user's interest to our keywords.

3 Experiment

In order to determine the feasibility of the proposed approach to deriving relationships between tags from folksonomies for the purposes of tag-based user modeling and to determine the optimal setting of the algorithm, we performed several experiments with two different folksonomies. Our main concern was whether the algorithm creates cohesive groups of tags (subtrees) without significant flaws of the context.

3.1 Data

We collected a part of delicious bookmarking site (`http://delicious.com`) dataset by periodically polling their RSS feeds. First, we used a 'recent activity' RSS feed to obtain a list of 128 448 unique user login names, next we used this information in user-scoped RSS feeds to obtain all tags and tagged pages for a given login name.

As the second dataset, we took the anonymized folksonomy of users-tags-publications from the CiteULike (`http://citeulike.org`), which is a system for tagging and searching for scholarly papers. Summarization of data we were able to acquire so far is listed in the Table 1.

First thing we were interested in was whether these two folksonomies are used in a similar manner. The graph on Fig. 1 depicts a distribution of tags on pages in delicious in a logarithmic scale. We can see that the distribution of tags

[1] hypernym – the generic term used to designate a whole class of specific instances. Y is a hypernym of X if X is a (kind of) Y

Table 1. Overview of the acquired dataset.

	delicious	citeulike
#usernames:	128 448	44 215
#records:	2 957 144	5 228 356
#processed users:	2 234	44 215
#unique tags:	220 647	294 806
#unique items(pages,publications):	962 367	1 437 245

fits more or less the power-law, with 421 840 pages tagged by one tag only and one page having 171 distinct tags[2].

Similar graph for the publications of the CiteULike dataset is shown on Fig. 2. Again, we see the power law distribution, with 625 658 publications tagged by one tag only and one publication tagged by 1 708 distinct tags[3].

We know already that power law distributions tend to arise in social systems where many people express their preferences among many options. Therefore, by observing the power law in both datasets, we were assured that datasets are valid and contain enough users (which could be an issue especially for our delicious dataset) to perform a crowd-based analysis. However, we found an interesting difference in popularity of tags. In our delicious dataset, we can find many tags which are shared among 5+% of all users whereas in the CiteULike dataset, a tag marked as "most active" on CiteULike website reaches popularity of 1 to 2 percents only. We continue in delicious feeds harvesting in order to determine, whether the overall popularity of tags marked as popular in our current dataset will decrease.

3.2 Results

We executed the algorithm 1 (see section 2) several times on both folksonomies with different setups and were observing the resulting trees. The setups of the variables for different runs of the algorithm are summarized in the Table 2. The *floating average overlap threshold* from the table means that the actual threshold is computed "on-the-fly" as a fraction of current parent-child overlap.

The manual inspection of resulting hierarchies proved the viability of our approach, where meaningful relations between keywords were created, which would definitely provide a good basis for a tag-based user modeling. The quality of the result depends highly on the configuration of the algorithm. The rest of this section is devoted to analysis of the impact of algorithm's variables to the created hierarchy.

We provide summarization of basic attributes of the produced hierarchies in the Table 3, with examples on Fig. 3 and 4 (complete results can be seen

[2] that page is `http://www.scribd.com/`

[3] the most tagged publication according to the CiteULike linkout database is surprisingly "about:blank"

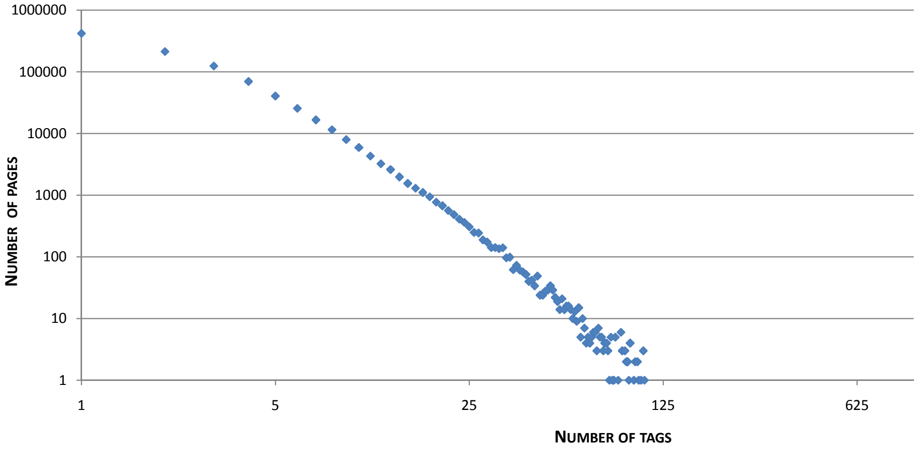

Fig. 1. Distribution of tags on pages in delicious dataset.

Table 2. Overview of algorithm setup.

	deli[1]	deli[2]	deli[3]	cite[1]	cite[2]	cite[3]
popularity threshold:	5%	5%	5%	1,5%	0,5%	0,5%
overlap threshold:	10%	5%	5%	10%	5%	5%
average overlap (context) threshold:	6,6%	3,3%	floating	6,6%	3,3%	floating

at `www.fiit.stuba.sk/~barla/iccci09`). Our taxonomies conform roughly to the criteria on estimating quality of a taxonomy defined in [9], where a high-quality taxonomy should have an average depth of 3 with a maximum depth of 5. On the delicious[1] and delicious[2] datasets, we can observe the impact of the overlap threshold. When set to 10% (delicious[1]), the tagsonomy tends to be more flat with highly consistent subtrees. The 5% setup of delicious[2] led to organizing more tags into subtrees (i.e., less tags on the first level), but showed that the contextual threshold 3,3% was setup weakly (only four tags were duplicated in order to keep up with context). The best results were achieved when contextual threshold was set as floating, according to current parent-child overlap. For instance, tag *currency* on the Fig. 3 which was wrongly assigned in a branch *audio/conversion* was moved into a separate subtree in order to keep the average overlap of tags in the *audio* branch on the higher level.

We were surprised by results coming from the CiteULike folksonomy. The first setup with 10% overlap threshold produced very small hierarchy with only 67 popular tags. It seems that in CiteULike, the crowd did not make an agreement on most appropriate tags for particular publication (i.e., everybody uses his or her own specific tags). One reason for such a difference could be that delicious is pro-actively supporting such an agreement to emerge by recommending tags

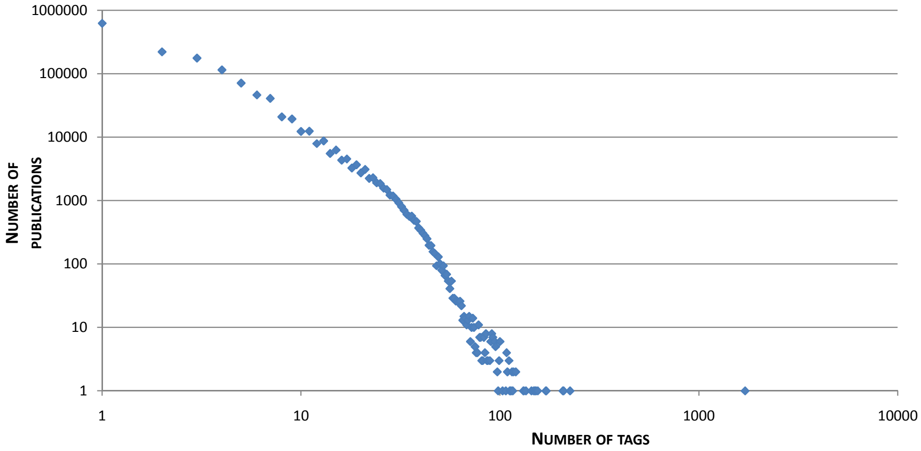

Fig. 2. Distribution of tags on publications in CiteULike dataset.

Table 3. Attributes of the resulting hierarchies.

	deli[1]	deli[2]	deli[3]	cite[1]	cite[2]	cite[3]
#popular tags:	1085	1085	1085	67	505	505
#duplicated tags:	1	4	94	0	5	25
#first-level tags:	357	172	277	54	83	125
#child-less first-lev. tags:	245	84	92	47	38	51
average depth:	1.8194	2.2074	2.0628	1.194	2.1743	1.9539
maximum depth:	5	5	5	2	5	5

when users are adding a particular page already present in the system, while CiteULike does not provide such a feature yet. Therefore, we decreased the required popularity to 0,5% for the CiteULike dataset, which resulted in 505 organized tags.

Resulting taxonomies pointed out yet another interesting difference between delicious and CiteULike folksonomies: CiteULike folksonomy contains words considered as English stop-words (and, on, the, for etc.) and (moreover) these stop-words are *popular*. This was something we did not expect at all, that somebody would use stop-words to organize some content. A possible explanation is that CiteULike users tend to post a short sentence as one tag (e.g., “example of a graph analysis”), but the CiteULike system considers each word of a sentence as a tag.

Another phenomenom of the CiteULike dataset is a popular *no-tag* tag, which is assigned automatically by the system if user enters a publication without any tags. Obviously, many people do use CiteULike without taking advantage of tagging system built in it, which was rather surprising finding.

```
3d                audio              blog
  modeling          sound              blogger
  blender           voice              blogging
  uml               audiobooks           comments
  opengl            recording          tumblr
                    sounds

3d                audio              blog
  model             conversion         bookmarks
  modeling            currency         del.icio.us
  virtual           convert            tag
  secondlife        converter            delicious
  blender           skype              internet
  uml               voip               telephone
  opengl            speech             domain
                    ...              security
                                         privacy
                                           rights
                                           human
                                       ...
```

Fig. 3. Example parts of tagsonomy created from the delicious dataset. The first row contains examples from the deli[1] setup, the second row from the deli[2] setup of the algorithm.

```
social            analysis           learning
  trust             pattern            machine
  community           matching         knowledge
  identity          reference            representation
  communities       performance        management
  virtual           recognition          strategy
  folksonomy          object           memory
  tagging           mass                 sleep
                    distribution       online
                    density            children
                    monitoring         problem
                    regression         plasticity
                    estimation
```

Fig. 4. Example parts of tagsonomy created from the CiteULike dataset using the cite[2] setup of the algorithm.

4 Related Works

The raise of the tagging systems naturally provoked increase of interest in analysis of folksonomy data among researchers. We are aware of various approaches in deriving additional knowledge from folksonomies for different purposes.

In [6], Mika presents two approaches to retrieve relationships between tags: concept-mining based on graph clustering algorithms (λ-set analysis) and set theory assumptions, which are similar to our work, but does not take into account contextual conditions nor popularity of the tags. It seems that it was performing acceptably on a small chosen domain (such as keywords related to semantic web).

Schmitz et al. [10] opted for association rules mining to build-up conceptual structure. Resulting rules have the form *Users assigning the tags from A to some resources often also assign the tags from B to them.* Even if authors did not provide the way how to derive a taxonomy from the mined rules, we can just look on them as on the subsumption relations, which we are deriving from the overlapping sets of tagged resources.

Heymann and Garcia-Molina [11] proposed another approach based on comparing tag vectors and connecting similar tags together. Then, the taxonomy is created according to tags' centrality measure in the created similarity graph.

Schwarzkopf et al. [9] extend both algorithms [10, 11] by taking into account a context of a tag, defined similarly to our work. They did not try to filter-out non-popular tags as we do, in order to obtain only a "crowd-agreement" tags, nor they do any further processing in order to enrich furthermore the taxonomy.

Shepitsen et al. [12] propose context-dependent hierarchical agglomerative clustering technique to organize tags into clusters subsequently used for recommendation of resources. As with any other clustering technique, a crucial part is the definition of similarity between items being clustered. Shepitsen et al. used cosine similarity of vectors over the set of tags.

We see several differences of our approach compared to the aforementioned ones. Apart from popularity, configurable bidirectional context checking (to ancestor or to children) and siblings detection, we proposed also an incorporation of other techniques and approaches into one corpus such as spreading activation, which greatly improves the resulting hierarchy and broaden its possible usage.

5 Conclusions

In this paper, we have shown a method how wisdom of the masses in the form of social bookmarking folksonomy can be used to create a "tagsonomy" (a taxonomy of tags). We also proposed other techniques with different background such as graph activation search coming from the graph theory and Wordnet's conceptual semantic relationships coming from the cognitive science area, which can contribute and enhance the final taxonomy of tags by adding new "shortcuts" between hierarchically ordered tags.

We performed several experiments with the algorithm on CiteULike and delicious folksonomies, which proved the viability of the approach and pointed out some interesting differences in the two mentioned tagging systems. We have shown that our algorithm for deriving hierarchy from the folksonomy can handle such differences when properly configured. More, the results proved that web 2.0 generated folksonomies can be used, when taking into account tags with a certain level of popularity optionally enriched by Wordnet's synonyms, for user's interest modeling for personalizing and adapting the web (and web 2.0 itself).

In our future work, we plan to evaluate the impact of spreading activation on relationships between tags and compare two different spreading methods (via actors, via items). Our preliminary results showed that the spreading algorithm is able to create meaningful "jumps" in the hierarchy, but is highly dependent on number of links from and to tag (when there are too many links, the energy is divided into neglectable chunks). One way how to solve this issue is to introduce a variable starting energy, depending on a popularity of a given tag.

Acknowledgment. This work was partially supported by the Scientific Grant Agency of Slovak Republic, grant No. VG1/0508/09 and the Cultural and Educational Grant Agency of the Slovak Republic, grant No. KEGA 3/5187/07.

References

1. Heckmann, D., et al.: GUMO – The General User Model Ontology. In Ardissono, L., et al., eds.: User Modeling 2005. LNCS 3538, Springer (2005) 428–432
2. Andrejko, A., Barla, M., Bieliková, M.: Ontology-based User Modeling for Web-based Information Systems. In: Advances in Information Systems Development. Springer (2007) 457–468
3. Barla, M., Tvarožek, M., Bieliková, M.: Rule-Based User Characteristics Acquisition from Logs With Semantics for Personalized Web-based Systems. Computing and Informatics (2009) accepted.
4. Coyle, M., Smyth, B.: (Web Search)shared: Social Aspects of a Collaborative, Community-Based Search Network. In Nejdl, W., Kay, J., Pu, P., Herder, E., eds.: AH 2008. LNCS 5149, Springer (2008) 103–112
5. Joachims, T., et al.: Accurately interpreting clickthrough data as implicit feedback. In: SIGIR 2005, ACM (2005) 154–161
6. Mika, P.: Ontologies are us: A Unified Model of Social Networks and Semantics. J. Web Sem. **5**(1) (2007) 5–15
7. Crestani, F.: Application of Spreading Activation Techniques in Information Retrieval. Artif. Intell. Rev. **11**(6) (1997) 453–482
8. Fellbaum, C.: WordNet: An Electronical Lexical Database. The MIT Press, Cambridge, MA (1998)
9. Schwarzkopf, E., Heckmann, D., Dengler, D., Kröner, A.: Mining the Structure of Tag Spaces for User Modeling. In: Data Mining for User Modeling, Workshop held at UM2007. (2007) 63–75
10. Schmitz, C., et al.: Mining Association Rules in Folksonomies. Studies in Classification, Data Analysis, and Knowledge Organization, Part VI. In: Data Science and Classification. Springer, Berlin Heidelberg (2006) 261–270
11. Heymann, P., Garcia-Molina, H.: Collaborative Creation of Communal Hierarchical Taxonomies in Social Tagging Systems. Technical report, Computer Science Department, Stanford University (2006) Available at: http://heymann.stanford.edu/taghierarchy.html (29.06.2009).
12. Shepitsen, A., et al.: Personalized Recommendation in Social Tagging Systems using Hierarchical Clustering. In: Recommender Systems 2008, New York, NY, USA, ACM (2008) 259–266