This is the author’s version of a work that was submitted/accepted for publication in the following source:


This file was downloaded from: http://eprints.qut.edu.au/46776/

© Copyright 2011 [please consult the author]

Notice: Changes introduced as a result of publishing processes such as copy-editing and formatting may not be reflected in this document. For a definitive version of this work, please refer to the published source:
CONSTRUCTING TAG ONTOLOGY FROM
FOLKSONOMY BASED ON WORDNET

Endang Djuana, Yue Xu, Yuefeng Li
Faculty of Science and Technology
Queensland University of Technology
Brisbane, Australia
{e.djuanatjhwa, yue.xu, y2.li}@qut.edu.au

ABSTRACT
With the emergence of Web 2.0, Web users can classify Web items of their interest by using tags. Tags reflect users’ understanding to the items collected in each tag. Exploring user tagging behavior provides a promising way to understand users’ information needs. However, free and relatively uncontrolled vocabulary has its drawback in terms of lack of standardization and semantic ambiguity. Moreover, the relationships among tags have not been explored even there exist rich relationships among tags which could provide valuable information for us to better understand users. In this paper, we propose a novel approach to construct tag ontology based on the widely used general ontology WordNet to capture the semantics and the structural relationships of tags. Ambiguity of tags is a challenging problem to deal with in order to construct high quality tag ontology. We propose strategies to find the semantic meanings of tags and a strategy to disambiguate the semantics of tags based on the opinion of WordNet lexicographers. In order to evaluate the usefulness of the constructed tag ontology, in this paper we apply the extracted tag ontology in a tag recommendation experiment. We believe this is the first application of tag ontology for recommendation making. The initial result shows that by using the tag ontology to re-rank the recommended tags, the accuracy of the tag recommendation can be improved.

KEYWORDS
Collaborative tagging, Ontology learning, Tag recommendation

1. INTRODUCTION
Collaborative tagging describes the process by which users add metadata in the form of keywords to Internet resources with a freely chosen set of keywords (tags) (Marlow et al. 2006, Golder and Huberman 2006). Social tagging systems allow users to share their tags for a particular resource (Marlow et al. 2006). Recently, user tagging systems have grown in popularity on the web that allows users to tag bookmarks, photographs and other content. These can be found in e-commerce sites such as Amazon, social bookmarking tools such as Delicious, photo-sharing sites such as Flickr. Similar concept to tagging is also found in micro blog sites such as Twitter, in the form of “hashtag” for describing topic of discussions and “list” for organizing users’ posts into personalized grouping. The tagging process is quite simple for an ordinary user who doesn’t need to have systematic classification background which brought to its popularity. However, free and relatively uncontrolled vocabulary has its drawback in terms of lack of standardization and semantic ambiguity. Three of these problems are polysemy, synonymy, and basic level variation (Golder and Huberman 2006). Also, the flat and non-hierarchical structure leads to low search precision and poor resource navigation.

Collaborative tagging systems usually include tag recommendation mechanism to assist with the process of finding good tags for an item. The task of a tag recommender system is to recommend for a given user and a given item a set of tags for annotating the item. It typically generates a set of tags usually ranked based on some quality of relevance criterion from which the top ranked tags are selected and recommended to users (Jaschke et al. 2008). Despite the different approaches taken, tag recommender systems have to overcome inherent problems with user tagging information which are the semantic ambiguity and the lack of hierarchy among terms used for tagging an item. To be able to recommend the most relevant tag for a resource to a user, the semantic meaning of tags used by users especially the semantic relationships between tags in the tag collection should be taken into consideration to generate the tag recommendations. However, semantic
relationships between tags have not been sufficiently exploited in the existing proposed methods. These problems motivate the work that we introduce in this paper that aims to represent the semantic meaning and relationship of tags for the purpose of making recommendation.

We believe that having ontology of tagging information in place can benefit tag recommendation process. In this paper we present our approach to ontology learning from user tagging information based on the widely used general ontology WordNet. We begin by providing a bit of backgrounds of user tagging collection and tag recommendation process in Section 2. We then introduce our ontology learning approach in Section 3. In Section 4 we discuss the evaluation methods, pilot experiment and the initial results. In Section 5 we review related works and compare our approach to others. Section 6 concludes this paper and gives some ideas for further work.

2. BACKGROUND

In this section we provide background of user tagging collections and the tag recommendation process.

2.1 User Tagging Collection

User tagging collection consists of three entities which are items, tags assigned to these items and users who assign these tags to the items. Those three entities are described as follows:

- **Users** \( U = \{ u_1, u_2, \ldots, u_k \} \): contain all users in an online community who have used tags to organize their items. Users are typically described by their user ID.

- **Tags** \( T = \{ t_1, t_2, \ldots, t_k \} \): contain all tags used by the users in \( U \). Tags are typically arbitrary strings (which could be a single word or short phrase).

- **Items** \( I = \{ i_1, i_2, \ldots, i_k \} \): contain all domain-relevant items or resources, for instance, in Amazon the items are mainly books.

Based on the three entities of user tagging collection, the collaborative tagging system is formulated as 4-tuple: \( (U, T, I, Y) \) by Jaschke et al. (2008) where \( U, T, I \) are finite sets, whose elements are the users, tags and items respectively. \( Y \) is a ternary relation between them, i.e., \( Y \subseteq U \times T \times I \). An element \( (u, t, i) \in Y \) represents that user \( u \) collected item \( i \) using tag \( t \). A function \( F(u, i) \) is defined to return a set of tags that a user \( u \) has assigned to an item \( i \):

\[
F(u, i) = \{ t \in T \mid (u, t, i) \in Y \}
\]

for all \( u \in U \) and \( i \in I \).

2.2 Tag Characteristics and Challenges

Tags in a tag collection may exhibit many variations such as synonymy where multiple tags have the same or closely related meanings. In this situation items collected under synonymous tag are supposed to be collected under a consolidated tag rather than separate tags since they have similar meaning. The other variation is polysemy where one tag has multiple meanings. This condition causes bigger problem because although many items may be collected under particular tag, the items are not related to each other because they mean differently to users who annotate those items with this tag. Besides those variations mentioned above, one tag may have semantic relationship to other tags, e.g. “inn” is a kind of “hotel” which shows “more specific” and “more general” meaning. This condition may not be utilized to relate items collected under these two tags because they are simply treated as two different tags. Many methods have been proposed to deal with the problems of synonymy and polysemy (Bischoff et al. 2008, Liang et al. 2010, Suchanek et al. 2008). However, the semantic relationship between tags has not been exploited by existing tagging based applications including tag based recommender systems.

In order to tackle these problems, it becomes desirable to find a way to consolidate the multiple facets (i.e., different meanings) and the relationships of tags into a consolidated entity which will help better understand
the tags used by users. There are several possible solutions include using classification systems such as taxonomy or using conceptualization systems such as ontology. In this work we consider to use ontology to represent the semantics in tags collection rather than taxonomy because of the flexibility of an ontology and possibility of emerging semantics from the ontology learning process (Mika 2007, Navigli et al. 2003).

### 2.3 Tag Recommendation

A tag recommender is a specific kind of recommender systems in which the goal is to suggest a set of tags to use for a particular item to a user during the annotation process. The tags suggested are usually ranked based on some quality or relevance criterion from which the top ranked tags are selected. Based on previous formulation of collaborative tagging system the task of a tag recommender system is to recommend, for a given user $U \in U$ and a given item $I \in I$ with $F(u,i) = \emptyset$, a set $\tilde{T}(u,i) \subseteq T$ of tags. In many cases $\tilde{T}(u,i)$ is computed by first generating a ranking on the set of tags according to some criterion, for instance by a collaborative filtering, content based, or other recommendation algorithms, from which then the top $n$ tags are selected (Jaschke et al. 2008).

Given this task of producing a set of tags based on some ranking criteria, one of our immediate goals was whether the semantic relationship among tags can be utilized to improve the ranking calculation for a set of candidate tags among other possible improvements. Here we argue that utilizing the tag ontology to generate the candidate tags and calculate the ranking may improve the accuracy of the tag recommendation results.

### 3. ONTOLOGY LEARNING FROM USER TAGGING

How to construct ontology is one challenging problem as manually identifying, defining and entering concept definition can be a lengthy and costly process. One stream of approach to the ontology construction relies on machine learning and automated language-processing techniques to extract concepts and ontological relations from structured and unstructured data such as database and text (Navigli et al. 2003). In this work we propose to construct the tag ontology based on foundational ontology, which we call backbone ontology to map the tags in the tag collection to the concepts on the backbone ontology and make use of the available relationships among concepts in the backbone ontology. We chose WordNet general ontology (Felbaum 1998) as the backbone ontology as it has wide coverage of concepts (over 200,000) and richness of relationships such as semantic relationships “is-a”, “part-of”, lexical relationships “synonymy” and “antonymy” as well as availability of accompanying corpus and other facility for disambiguation process.

Two main tasks are included in the proposed tag ontology construction: to find the meaning of user tags and to find the relationships among tags. Accordingly, there are two stages in the proposed approach, mapping tags to the concepts in the backbone ontology and extracting the relations between the mapped tags from the backbone ontology. As mentioned before, a tag may have multiple meanings, which causes potential multiple mappings. Therefore, for the first stage, disambiguation is needed to identify the most relevant concept for a tag. The second stage involves finding all the links between the mapped concepts by going through the hierarchy in the backbone ontology for semantic relationships such as “is-a” or “part-of”. In order for us to discuss our proposed approach in more detail we will first give a formal definition to ontology, and then we introduce the mapping disambiguation process and the relation extraction in the following sub sections.

### 3.1 Ontology Definition

In this section, we will define the backbone ontology first before defining other relevant concepts.

**Definition 1 (Backbone ontology)**: The backbone ontology is formally defined as a 2-tuple $\text{BackboneOnto} = (C, R)$ where $C = \{c_1, c_2, \ldots, c_{|C|}\}$ is a set of concepts; $R = \{r_1, r_2, \ldots, r_{|R|}\}$ is a set of relations representing the relationships between concepts.

**Definition 2 (Concept)**: A concept $c$ in $C$ is a 4-tuple $c = (id, synset, gloss, category)$ where $id$ is a unique identification assigned by WordNet system to the concept $c$; $synset$ is a synonym set containing synonymic
terms which represent the meaning of the concept \( C \); \textit{gloss} is a short definition in natural language describing the meaning of the concept \( C \); and \textit{category} is a lexical category assigned by WordNet lexicographer to classify this concept \( C \) into a general category.

For easy to describe the work, we denote the identifier of a concept \( C \) by \( \text{id}(C) \), the set of synonyms representing \( C \) by \( \text{synset}(C) \), the gloss of \( C \) by \( \text{gloss}(C) \) and the category of \( C \) by \( \text{category}(C) \).

Let \( S = \{ w \mid \exists c \in C, w \in \text{synset}(c) \} \) be the set of all synonomic terms, for a term \( w \in S \), the set of concepts for which \( w \) is a synonymic term is defined as \( \text{con}(w) = \{ c \mid w \in \text{synset}(c) \} \). For the terms in a \( \text{synset} \), each term \( w \in \text{synset}(c) \) is a 2-tuple \((w, \text{freq}_c(w))\), where \( w \) is a synonym in the \( \text{synset} \); \( \text{freq}_c(w) \) is the frequency assigned by WordNet to the term as an indication of how frequently this term has been used to represent the meaning of the concept based on the accompanying WordNet corpus.

**Definition 3 (Relation):** A relation \( r \) in the relation set \( R \) is a 3-tuple \( r = (\text{type}, x, y) \), where \( \text{type} \in \{ \text{isA}, \text{partOf} \} \); \( x, y \) are the concepts that hold the relation \( r \).

### 3.2 Identifying Semantic Meaning of Tags

One tag may contain one or more terms. It is possible that a tag can be mapped directly to one or more concepts in the backbone ontology. It is also possible that only part of a tag may map to one or more concepts. We propose the following mappings to deal with different cases. There are 3 different cases for finding possible mappings for a given tag, which are: 1) mapping the full tag to one or more concepts; 2) mapping part of the tag to one or more concepts; and 3) splitting the tag into a list of single words, then mapping each of the words to concepts separately. We will describe each case in the following discussion.

1) **Direct Mapping**

First of all, for each tag, we try to map the tag as a whole to the concepts in the backbone ontology. In this paper, if the tag is a synset term of a concept, the concept is considered a mapping of the tag. We define the following function to represent the mapping from a tag to concepts:

\[
\text{TagConcept}: T \rightarrow 2^C, \\forall t \in T, \text{TagConcept}(t) = \{ c \mid c \in C, \exists (w, f) \in \text{synset}(c), t \equiv w \}
\]

On the other hand, the tag \( t \) is also considered a set of concepts for each of which \( c \) is a synset term.

The following function defines the mapping from a concept to tags:

\[
\text{ConceptTag}: C \rightarrow 2^T, \\forall c \in C, \text{ConceptTag}(c) = \{ t \mid t \in T, \exists (w, f) \in \text{synset}(c), t \equiv w \}
\]

2) **Partial Mapping**

We observed that most tags that can’t be directly mapped to any concepts in the WordNet ontology are phrases. Especially, we found that most of the phrases have their head word placed at the end of the phrase with some modifiers appearing before the head word, for example, “2nd world war”. We could not find a mapping for this phrase, but we can find a mapping for “world war”. Based on this observation, we propose to use the mapping of the largest partial phrase as the mapping of a phrase if the phrase can’t map to any concept.

It could be an option to check all the possible partial phrases to find a mapping. According to our observation the head word is often placed at the end of the phrase. In the current work, we only check postfix partial phrases for a mapping. That is, if a tag can’t map to any concept as a whole, only the largest postfix of the tag is checked to find the mapping of the tag. For a tag \( t = \langle \text{term}_1, \text{term}_2, ..., \text{term}_m \rangle \), a partial tag \( t_i = \langle \text{term}_i, ..., \text{term}_m \rangle \) is considered a largest mappable postfix of \( t \) if \( t_i \) can map to a concept and any of its partial tags \( \langle \text{term}_i, ..., \text{term}_m \rangle \) can’t be mapped to any concept where \( 0 < i < k - 1 \) and \( k \geq 2 \). To find the largest postfix of a tag, its partial tag with the left-most term removed is checked to see if
the partial tag is mappable until a mapping is found or the end of the phrase is reached which indicates that there is no partial mapping to this tag.

3) Term Mapping

For each of the remaining tags, we conducted the split tag mapping. The function $\text{tagset}(t)$ defined in section 2.1 returns a set of individual terms that make up the tag $t$. We first map each of the terms to a concept, then conduct a disambiguation process which will be discussed in the next subsection to determine which of the mapped concepts should be chosen to be the mapping of this tag.

After the mapping process discussed above, not only the mapped concepts of the tags but also the strength of the mappings were generated. The strength of a mapping is measured by the word frequency associated with the synset term of the mapped concept. A matrix $\text{TagCon} [t, c]_{mn}$ called Tag Concept matrix is defined to represent the strength of the mapping between tags and concepts, where $m=|T|$ and $n=|C|$. The initial mapping strength is defined as below:

$$\text{TagCon} [t, c] = \begin{cases} \text{freq}_{c}(t) & c \in \text{TagConcept}(t) \\ 0 & \text{otherwise} \end{cases}$$

(1)

$\text{TagCon} [t, c]$ in the matrix represents the mapping strength between $t$ and $c$ based on the statistics from the WordNet corpus. For example, assume that tag $t_1$ in Figure 1 below are mapped to two concepts $c_1$ and $c_2$ with term frequency $\text{freq}_{c_1}(t_1)=6$, $\text{freq}_{c_2}(t_1)=1$, then the mapping strength from $t_1$ to the two concepts is $\text{TagCon} [t_1, c_1]=6$ and $\text{TagCon} [t_1, c_2]=1$, respectively. If the value is 0, it indicates that there is no mapping between the tag and the concept.

A tag in the tag collection may map to multiple concepts in the backbone ontology which indicates that there exists ambiguity in the interpretation of the tag. Figure 1 illustrates the relationships between concepts, tags and items. For example, tag $t_1$ in Figure 1 is used to tag two items $a_1$ and $a_2$, and mapped to two concepts $c_1$ and $c_2$.

![Figure 1 An Example of Relationships between Tags, Items, and Concepts in Tagging](image_url)

After all the possible mapped concepts are found for a tag, we need to choose the most appropriate concept from the mapped concepts to represent the meaning of the tag for this particular tag collection. A strategy is proposed to disambiguate the concepts of a tag which makes use of the synset term frequency of the concepts.

For a tag $t$ and a set of concepts: $\text{TagConcept}(t)=\{c_1, c_2, ..., c_p\}$, as defined in Equation (1), $\text{TagCon}[t, c], i=1,2, ..., p$, is the term frequency of the term $t$ to represent the concept $c_i$. In order to make the frequency comparable between different concepts and terms, we normalize the frequency value to a scale of $[0, 1]$. Moreover, as $\text{TagCon}[t, c]$ may be a subset of $\text{con}(t)$, i.e., $\text{TagCon}[t, c] \subseteq \text{con}(t)$, in order to more accurately represent the mapping strength from $t$ to the concepts in $\text{TagConcept}(t)$ only, the term frequencies are normalized over the concepts in $\text{TagConcept}(t)$ instead of all the concepts in $\text{con}(t)$.

Equation (1) is modified as below in Equation (2) to generate a matrix $\text{TagCon} \text{frequency} [t, c]_{mn}$ which provides the normalized frequency instead of the original term frequency. For a tag $t$, $c_j$ should be chosen as $t_i$’s concept if $\text{TagCon} \text{frequency} [t, c_j]$ is the highest value for all $c_j \in \text{TagConcept}(t_i)$. For the example
mentioned in Figure 1, the mapping strength from tag $t_1$ to concepts $c_1$ and $c_2$ becomes $TagCon_{frequency}[t_1, c_1] = 6/7$ and $TagCon_{frequency}[t_1, c_2] = 1/7$, respectively.

$$TagCon_{frequency}[t_1, c_j] = \begin{cases} \frac{\text{freq}_{c_j}(t_1)}{\sum_{c_j \in \text{TagConcept}(t_1)} \text{freq}_{c_j}(t_1)} & c_j \in \text{TagConcept}(t_1) \\ 0 & \text{otherwise} \end{cases}$$ \hspace{1cm} (2)

3.3 Relationship Extraction Process

Initially in the mapping process after we collect all the available tag-to-concept mapping: $TagCon[t_1, c]$, we retrieve the available “is-a” and “part-of” relation from the mapped concept consecutively until we reach the top of the hierarchy. This operation is the same operation as finding an ancestor in a tree-based structure. The top of the hierarchy in the backbone ontology is a general category defined by WordNet as described in Definition 2. We can then extract the mapped concepts together with the relationships in the backbone ontology to form the tag ontology. As the result of the tag to concept mapping and the relationships extraction, we can construct the tag ontology which is defined as below:

**Definition 4 (Tag Ontology):** The tag ontology is defined as 2-tuple $TagOnto = (TC, TR)$ where $TC = \{tc_1, tc_2, ..., tc_{|TC|}\}$ is a set of tag-concepts and $TR = \{tr_1, tr_2, ..., tr_{|TR|}\}$ is a set of tag relations.

**Definition 5 (Tag Concept):** The tag-concept $TC$ in the tag ontology is defined as $TC \subseteq C \times 2^T$. Each element in TC is a pair of a concept $c$ and a set of tags $\{t_1, t_2, ..., t_n\}$, i.e., $tc = (c, \{t_1, t_2, ..., t_n\}) \in TC$, which represents that each tag $t \in \{t_1, ..., t_n\}$ can be mapped to the concept $c$.

**Definition 6 (Tag Relation):** The tag relation $TR$ in the tag ontology is defined as:

$$TR = \{r = (type, c_1, c_2) \mid r \in R, ConceptTag(c_1) \neq \emptyset, ConceptTag(c_2) \neq \emptyset\}$$

which represents the subset of all relations between concepts in the backbone ontology. An element $tr \in TR$ is the extracted relation between tag concepts.

4. EVALUATION

4.1 Baseline System

In order to evaluate the potential improvement to tag recommendation process, we implement a baseline tag recommender system proposed in (Jaschke et al. 2008) which is based on the user-based collaborative filtering (CF) method. The recommendation result of this baseline system is then modified based on the tag relation information obtained from the tag ontology proposed in this paper. The original recommendations and the modified recommendations are compared to indicate the improvement achieved by using the proposed tag ontology.

The baseline user-based CF tag recommender aims to generate a set of tags which are ranked based on tags used by other users to tag a particular item that a target user is concerned. The ranking calculation conducted may result in a tie. Ties between ranking values of tags are often solved by random selection. This leads to a potential problem of leading to a good tag being missed out due to the random selection process.

4.2 Proposed Improvement

Based on the constructed tag ontology, we have explored for a possible improvement to the potential ranking tie problem mentioned above. This brought us an idea to propose a re-ranking approach based on semantic relations in the extracted ontology to see if the ontology can directly improve tag recommendations. The goal of the re-ranking is to re-order the recommended tag list such that highly relevant tags are encouraged to move upwards. Our Re-ranking approach compares the relative distance between the recommended tags to determine if one tag is more specific or more general in terms of hierarchy. The further away from the top of
hierarchy, the more specific a tag is. We assign a score based on this relative position to each tag. The more specific a tag, the higher the score is to this tag and the higher the score, the higher the rank is.

We have conducted experiments to evaluate the methods proposed in Section 3. Two datasets are used in the experiments. The first dataset is the publicly available Delicious dataset (Wetzler et al. 2008). The dataset contains all public bookmarks of users posted on delicious.com between September 2003 and December 2007. In this paper a portion of the data set is used which contains bookmarks from January to July 2004. This portion contains 8792 users, 318175 items (URLs) and 55508 tags. The total posts are 517411 bookmarks. To avoid sparsity problem, we selected those users who tagged at least 5 items, tags that appear in minimum 5 posts and items that are tagged at least 5 times. The second dataset is from a real world industry organization. This dataset contains 1000 anonymous users. The items in this dataset are posts about places of interest. To avoid severe sparsity problem, we selected those users who tagged at least 3 items, tags that are used by at least 3 users and items that are tagged at least 3 times.

Each of the datasets is split into a testing dataset and a training dataset based on posting date. The split percentage is 25% for testing dataset and 75% for training dataset. In the experiments we conducted 5 folds cross validation for all the users in the dataset. In each run of the experiment, we randomly take 20% portion as the target users while the remaining 80% is taken as the training users from whom we calculate similarities to the target users to find neighbors. The top $n$ tags are recommended to each target user for each of the user’s items in the testing set. The recommended tags are compared to the target user’s actual tags of the items in the testing dataset. If a recommended tag matches with an actual tag, we calculate this as a hit. The standard precision and recall are used to evaluate the accuracy of tag recommendations.

### 4.4 Results and Discussion

The results of the experiments are presented in Table 1 and 2 for Delicious dataset and in Table 3 and 4 for the real world dataset. As shown in Table 1 and 3, the re-ranking based on the position of the tags in the constructed tag ontology has improved the precision apparently for the Delicious dataset and for the real world dataset slightly. However, the improvement to recall is very slight except for the recommendation with size 3 or 5. From the results of this experiment, we can say that there-rank methods will improve the results the most for smaller recommendation size, which in most cases are more desirable since normally the tag recommender systems recommend less than top 10 tags to use.

<table>
<thead>
<tr>
<th>N</th>
<th>User-CF</th>
<th>Re-rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.07236</td>
<td>0.10231</td>
</tr>
<tr>
<td>10</td>
<td>0.07026</td>
<td>0.10230</td>
</tr>
<tr>
<td>15</td>
<td>0.06428</td>
<td>0.10230</td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>N</th>
<th>User-CF</th>
<th>Re-rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.08378</td>
<td>0.10605</td>
</tr>
<tr>
<td>10</td>
<td>0.10116</td>
<td>0.10622</td>
</tr>
<tr>
<td>15</td>
<td>0.10248</td>
<td>0.10648</td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 5. RELATED WORK

Work by Mika (2007) shows that emergent semantics in the form of lightweight ontology can be extracted from social tagging system del.icio.us by performing graph transformation and affiliation network analysis.

There are several works which tried to extract ontological structures from user tagging systems. Lin et al. (2009) extracted ontological structures by exploiting low support association rule mining supplemented by WordNet. Trabelsi et al. (2010) focused more on extracting non-taxonomic relationships from folksonomies using triadic concepts with external resources: WordNet, Wikipedia and Google.

Baruzzo et al. (2009) used existing domain ontology in software engineering to recommend new tags by analyzing textual content of a resource needed to be tagged. They relied on existing domain ontology which is
not always available for a particular domain and also in this work they didn’t provide quantitative evaluation as usually presented in other tag recommendation work.

Tatu et al. (2008) won the tag recommendation task in the 2008 ECML PKDD Discovery Challenge by mapping textual contents in Bibsonomy bookmarks, not just the tags assigned to a bookmark, to form what they called conflated tags to normalized concepts in WordNet. Although their approach is comprehensive, they relied on extended textual contents provided by Bibsonomy which are not always available in other user tagging systems.

6. CONCLUSION AND FURTHER WORK

Tagging is getting more and more popular in many Web sites. It provides useful data for better understanding users’ information needs. In this paper, we proposed a novel approach to construct tag ontology from user tagging information. We believe the constructed tag ontology can be used in many applications such as item classification, item recommendation, and tag recommendation. In this paper, we also presented a primary experiment to show the improvement to tag recommendation by re-ranking the recommendations based on the tag ontology. There is room to improve the recommendation by exploiting further the extracted ontology structure for instance by considering the distance among concepts to find more neighbors and reducing the sparsity problem.

REFERENCES


Pearl, J 1988, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, Los Altos, CA.


