Generating ontologies with basic level concepts from folksonomies

Chen, Wen-Hao; Cai, Yi; Leung, Ho-Fung; Li, Qing

Published in:
Procedia Computer Science

Published: 01/01/2010

Document Version:
Final Published version, also known as Publisher's PDF, Publisher's Final version or Version of Record

Published version (DOI):
10.1016/j.procs.2010.04.061

Publication details:
Chen, W-H., Cai, Y., Leung, H-F., & Li, Q. (2010). Generating ontologies with basic level concepts from folksonomies. *Procedia Computer Science, 1*(1), 573-581. https://doi.org/10.1016/j.procs.2010.04.061

Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 1 (2012) 573–581
Procedia Computer Science
www.elsevier.com/locate/procedia

International Conference on Computational Science, ICCS 2010

Generating ontologies with basic level concepts from folksonomies

Wen-hao Chen[a,1], Yi Cai[b], Ho-fung Leung[a], Qing Li[b]

[a]*Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, China*
[b]*Department of Computer Science, City University of Hong Kong, Hong Kong, China*

Abstract

This paper deals with the problem of ontology generation. Ontology plays an important role in knowledge representation, and it is an artifact describing a certain reality with specific vocabulary. Recently many researchers have realized that folksonomy is a potential knowledge source for generating ontologies. Although some results have already been reported on generating ontologies from folksonomies, most of them do not consider what a more acceptable and applicable ontology for users should be, nor do they take human thinking into consideration. Cognitive psychologists find that most human knowledge is represented by basic level concepts which is a family of concepts frequently used by people in daily life. Taking cognitive psychology into consideration, we propose a method to generate ontologies with basic level concepts from folksonomies. Using Open Directory Project (ODP) as the benchmark, we demonstrate that the ontology generated by our method is reasonable and consistent with human thinking.

Keywords: Folksonomy, Ontology, Basic Level Categories, Category Utility

1. Introduction

The goal of generating ontologies is to automatically extract relevant concepts and their relations from a certain data set. Since ontology plays an important role in providing a way to give semantics to web resource and it is a time consuming job for human to construct an ontology, research has been conducted on automatically extracting ontologies from texts and other resources such as folksonomies. Recently, folksonomies have become popular as part of social annotation systems such as social bookmarking (e.g., delicious.com[2]) and photograph annotation (e.g., flickr[3]). It provides an user-friendly interface for people to annotate web resources freely, and also enables users to share their annotations on the web. These annotations are known as folksonomy tags, which provide a potential source of user-created metadata. Al-Khalifa et al. [1] demonstrated that folksonomy tags agree more closely with human thinking than those automatically generated from texts. Ontologies constructed from these tags may directly represent users' opinions about how to describe web resources and be more easily accepted by others than the ontology generated from texts [2]. Previous research on ontology extraction from folksonomies focused on hierarchy construction of tags and

Email address: whchen@cse.cuhk.edu.hk (Wen-hao Chen)
[1]Corresponding author
[2]http://delicious.com
[3]http://www.flickr.com

1877-0509
doi:10.1016/j.procs.2010.04.061

lacked a principle for supervising the process from a human's perspective. Since an ontology provides a vocabulary shared by users to model a domain, it is necessary to construct ontologies from users' perspective (i.e., taking how people define and use concepts into consideration).

According to the studies of cognitive psychology, there is a family of categories named basic level categories [3] [4]. People most frequently prefer to use basic level concepts constructed from these categories in their daily life, and these concepts are the ones first named and understood by children. For example, when people see a dog, although we also can call it an "animal" or a "terrier", most people would call it a "dog". What is more, most human knowledge is represented by basic level concepts. Thus, we consider that it is more acceptable and applicable for users by constructing an ontology with basic level concepts. Experiments in this paper demonstrate the significance for such a consideration by comparing the generated ontologies with ODP [4] which are built by human users.

In this paper, we focus on generating ontologies with basic level concepts from folksonomies based on studies in cognitive psychology. To the best of our knowledge, it is the first work on discovering basic level concepts from folksonomies and using them to construct ontologies. We perform experiments to evaluate our method using delicious.com data set and compare the generated ontology with ODP concept hierarchy. Experiments show that the ontologies generated using our method are more consistent with human thinking than that of other compared methods. Figure 1 gives an example of the ontology explored through our approach. In our approach, concepts are represented by the common tags of a category of resources. For example, tags "java" and "programming" together represents a concept about java programming. The tags of a concept are inherited by its sub-concepts and a concept has all instances of its descendants. Such a representation can keep more information and properties of concepts and is consistent with definition of concepts in psychology.

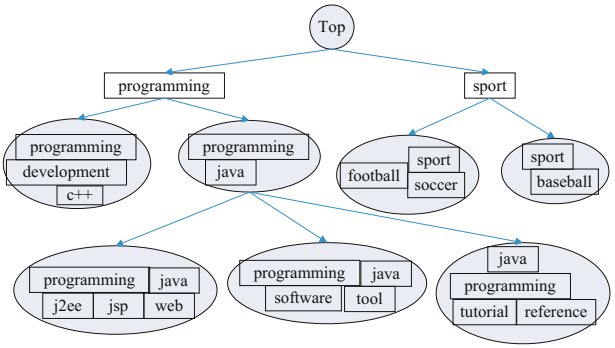

Figure 1: An ontology generated by our approach.

2. Preliminaries

2.1. Folksonomy

We use the definition of folksonomy given in [5]. In the definition, users are described by their user IDs, and tags are arbitrary strings. The type of resources in a folksonomy depends on the social annotation system,[5] and users create tags to annotate resources.

Definition 1. *A* **folksonomy** *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, respectively, and Y is a ternary relation over them, i.e.* $Y \subseteq U \times T \times R$.

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

[5] In delicious, for example, resources are web pages while in Flickr resources are images and videos.

2.2. Ontology

Ontology is a formal specification of conceptualization and an important component in the Semantic Web [6]. Generally, an ontology consists in a hierarchical taxonomy of concepts. Every concept consists of a category of instances and is described by its properties. The hierarchy of an ontology is indeed a taxonomic (subclass) hierarchy [7] [8]. Following [9], we define an ontology as follows:

Definition 2. *An* **Ontology** *is a tuple* $O = (C, P, I, S)$ *where C, P and I are finite sets, whose elements are called concepts, properties and instances, respectively, and S is a set of rules, propositions or axioms that specify the relations among concepts, properties and instances.*

2.3. Basic Level Categories and Basic Level Concepts

In cognitive psychology, in a hierarchical category structure such as a taxonomy of plants, there is one level named the basic level at which the categories are cognitively basic. The basic level categories, defined by Rosch et al. [3], carry the most information and are the most differentiated from one another. They are the categories easier than others to learn and recall by humans as concepts. In psychology, generally a concept holds the common features of a category of instances and is the abstraction of that category. Basic level concepts are the abstraction of basic level categories. Objects are identified as belonging to basic level categories and recognized as the basic level concepts faster than others. For example, in classifying life forms, basic level categories tend to be at the level of the genus (maple, dog etc.). If we see a maple, we could call it a "plant", a "maple" and a "sugar maple", but most people will identify it as "maple". The concept "maple" is a basic level concept.

3. Generating Ontologies with Basic Level Concepts from Folksonomies

3.1. Modeling Instances and Concepts in Folksonomies

In folksonomies, tags are given by users to annotate a resource and describe its characters. Naturally, the tagged resources are considered as instances in the definition of ontology. For the reason that each resource is described and represented by tags, we consider these tags as properties of instances. Accordingly, an instance is represented as a vector of tag-value pairs:

Definition 3. *An* **instance**, r_i, *is represented by a vector of tag:value pairs,* $r_i = (t_{i,1} : v_{i,1}, t_{i,2} : v_{i,2}, \ldots, t_{i,n} : v_{i,n})$ *with* $t_{i,k} \in T, 0 < v_{i,k} \leq 1, 1 \leq k \leq n$.

where n is the number of the unique tags assigned to resource r_i, $v_{i,k}$ is the weight of tag $t_{i,k}$ in resource r_i. The weight $v_{i,k}$ determines the importance of the tag $t_{i,k}$ to resource r_i. We consider that a tag assigned by more users to a resource is more important because more users think the tag is useful to describe the resource. Although different users may annotate a resource in different aspects and some may even randomly assign tags, Golder [10] demonstrated that, in delicious.com, in a resource the occurrence frequency of a tag becomes a nearly fixed number after enough bookmark. The fixed number reflects the importance of a tag in the resource. Accordingly the weight of a tag $t_{i,k}$ is defined as $v_{i,k} = \frac{N_{i,k}}{N_{r_i}}$, where $N_{t_{i,k}}$ is the number of users using the tag $t_{i,k}$ to annotate the resource r_i and N_{r_i} is the total number of users assigning tags to r_i. In the case that all users annotate r_i with $t_{i,k}$, the weight $v_{i,k}$ is 1.

A concept is the abstraction of a category of instances and holds the common properties of them. Accordingly, we construct a concept through extracting common tags of a category of instances. These common tags are considered as the properties of the concept. The weights of these tags are their mean values among all instances in a category. Accordingly the definition of a concept is as follows:

Definition 4. *A* **concept**, c_i, *is represented by a vector of tag:value pairs,* $c_i = (t_{i,1} : v_{i,1}, t_{i,2} : v_{i,2}, \ldots, t_{i,n} : v_{i,n})$ *with* $t_{i,k} \in T, 0 < v_{i,k} \leq 1, 1 \leq k \leq n$.

where n is the number of unique tags, $t_{i,k}$ is a common tag of a category of resources, $v_{i,k}$ is the weight of the tag $t_{i,k}$.

3.2. The Metric of Basic Level Categories

To characterize basic level categories, psychologists [11] give a metric named category utility. Through many experiments, they demonstrate that the character of basic level categories is that they have the highest category utility. Category utility was intended to supersede more limited measures of category goodness such as cue validity. It provides a normative information-theoretic measure of the predictive advantage gained by a person who possesses knowledge of the given category structure over a person who does not possess this knowledge. Given a set C of categories and a set F of features, the category utility is defined as follows:

$$cu(C,F) = \frac{1}{m}\sum_{k=1}^{m} p(c_k)\left[\sum_{i=1}^{n} p(f_i|c_k)^2 - \sum_{i=1}^{n} p(f_i)^2\right] \tag{1}$$

where $p(f_i|c_k)$ is the probability that a member of category c_k has the feature f_i, $p(c_k)$ is the probability that an instance belongs to category c_k, $p(f_i)$ is the probability that an instance has feature f_i, n is the total number of features, m is the total number of categories.

Features of instances are represented by tags in folksonomies. Accordingly, in the definition of category utility, the tag set T is used as the feature set F and a tag t_i is used as a feature f_i. As we model, the importance of tags are different in folksonomies. To take the differences of tag importance into account, we modify the definition and add the weight w_i of tag t_i into the definition:

$$cu(C,T) = \frac{1}{m}\sum_{k=1}^{m} p(c_k)\left[\frac{\sum_{i=1}^{n_k} w_i p(t_i|c_k)^2}{n_k} - \frac{\sum_{i=1}^{n} w_i p(t_i)^2}{n}\right] \tag{2}$$

where w_i is the weight of the tag t_i, n_k is the number of unique tags in cluster c_k, n is the number of all unique tags. To reflect the mean weight of a tag, w_i is defined as $w_i = \frac{1}{N_{t_i}}\sum_{j=1}^{N_{t_i}} v_{j,i}$, where N_{t_i} is the number of resources annotated by tag t_i and $v_{j,i}$ is the weight of the tag t_i in resource r_j. To differentiate it from the original definition, we consider it as the weighted category utility.

3.3. Basic Level Categories (and Concepts) Detection

Because basic level categories (and concepts) have the highest category utility, the problem of finding basic level categories (and concepts) becomes an optimization problem using category utility as the objective function. The value of category utility is influenced by the intra-category similarity which reflects the similarity among members of a category. Categories with higher intra-category similarity have higher value of category utility. Accordingly, we put the most similar instances together in every step of our method until the decrease of category utility. To compute the similarity, we use the idf-cosine coefficient [12] which is a commonly used method of computing similarity between two vectors in information retrieval. It is defined as follows:

$$sim(a,b) = \frac{\sum_{k=1}^{n} idf(t_k)\cdot v_{a,k}\cdot v_{b,k}}{\sqrt{\sum_{k=1}^{n} v_{a,k}^2}\cdot\sqrt{\sum_{k=1}^{n} v_{b,k}^2}} \tag{3}$$

where a, b are two concepts, n is the total number of unique tags describing them, and $v_{a,k}$ is the value of tag $t_{a,k}$ in concept a, if a does not have the tag, the value is 0. $idf(t_k)$ is the inverse document frequency of the tag t_k, $idf(t_k) = \log_N(\frac{N}{N_{t_k}})$, where N is the total number of resources and N_{t_k} is the number of resources annotated by tag t_k, $0 \le idf(t_k) \le 1$. When $idf(t_k)$ is 0, the tag t_k is assigned to all resources. In this case, all resources have this tag, the tag is not useful for categorization.

In our algorithm, firstly, we consider every single instance itself as a concept. This type of concept which only includes one instance is considered as the bottom level concepts. Secondly, we compute the similarity between each pair of concepts and build the similarity matrix. Thirdly, the most similar pair in the matrix is identified and merged into a new concept. The new concept contains all instances of the two old concepts and holds their common properties. After that we reconsider the similarity matrix of the remaining concepts. We apply this merging process until only one concept is left or the similarity between the most similar concepts is 0. We then determine the step where the categories

have the highest category utility which is the local optimum of category utility. These categories are considered as the basic level categories and the concepts are considered as the basic level concepts. The detail of this algorithm is given in algorithm 1, and the time complexity is $O(N^2 \log N)$ where N is the number of resources.

Algorithm 1 Basic Level Concepts Detection

1: Input: R, a set of instances (resources)
2: Initialize C, C is an n dimensions vector $C = (c_1, c_2, ..., c_n)$ where its element c_i is the bottom level concept. C_{size} is equal to the number of elements in C. Set sim[n][n] as the similarity matrix of C, $sim[i][j] = sim(c_i, c_j)$. $S = (s_1, s_2, ..., s_n)$, s_i is used to record the clustering result of step i.
3: Set $s_1 = C$, step=1,
4: **while** $C_{size} > 1$ **do**
5: step++
6: Find the most similar concepts in C and define a new concept include all instances of them.
7: Delete the most similar concepts from C, and add the new concept into C.
8: Update the similarity matrix.
9: $C_{size} = C_{size} - 1$
10: Record the result, $s_{step} = C$
11: Compute the category utility of this step cu_{step}
12: **end while**
13: Find the step with the highest category utility cu_{max}, define the record of this step s_{max} as the basic level categories.
14: Define the concept of each basic level category. The concept includes all instances of the category and the properties of the concept are the common features (tags) of the instances.
15: Output these concepts.

3.4. Ontology Generation

To build the ontology, we first generate a root concept including all instances in algorithm 2. After finding the basic level concepts with algorithm 1, we add the basic level concepts to the ontology as sub-concepts of the root. After several iterations, a cognitively basic ontology is built. The psychological character differentiates the ontology built through our method to the ontology built using methods proposed in previous ontology learning research. The detail of this ontology generation method is given in algorithm 2.

Algorithm 2 Ontology Generation

1: Input: Concept c
2: Use algorithm 1 to explore basic level concepts from instances in c.
3: **if** the size of $s_{max} > 1$ **then**
4: **for** every element c_i in s_{max} **do**
5: Set c_i as the sub-concept of c
6: Use algorithm 2 with input c_i.
7: **end for**
8: **end if**

4. Experiments

4.1. Data Set and Experiment Setup

Experiments are performed on three genres of real world data : PROGRAMMING LANGUAGE, SPORT and GAME. The PROGRAMMING LANGUAGE data set consists of 1087 resources. The SPORT data set consists of 552 resources. The GAME data set consists of 645 resources. These data sets are crawled from delicious.com. As Golder [10] demonstrated, in delicious.com, each tag's occurrence frequency become fixed after a resource is bookmarked 100 times. The fixed frequency reflects the importance of a tag. To make sure that the frequency is nearly fixed, the web pages in our data sets are the ones which are bookmarked more than 100 times in delicious.com. In addition, the web pages in our data sets must appear in both delicious.com and Open Directory Project (ODP)[6] because we use ODP as the gold standard to evaluate the experiment results. ODP is a user-maintained hierarchical web directory. Each directory in ODP has a label describing its name (e.g., "Arts" or "Python") and is a category of web pages. These categories in ODP are created, verified and edited by thousands of users. Accordingly, ODP is

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

Table 1: Statistics of the generated ontologies

Data Set	#Resources	#Tags	#Users	#Concpets	#Levels
PROGRAMMING LANGUAGE	1087	39475	57976	422	6
SPORT	552	18776	31741	273	5
GAME	645	20352	39224	313	5

considered as an expert-generated ontology. The label of each directory is the name of the concept and the web pages in the directory are considered as the instances of this concept. To derive the gold standard ontology from ODP, we first choose a concept in the hierarchy of ODP, i.e. "Programming Languages" and then include all its sub-concepts and their descendants into the ontology. Furthermore, to filter the noise tags, we preprocess each data set by (a) removing stop words and tags whose weight are less than a threshold; (b) down casing the tags.

4.2. Quantitative Analysis

Table 1 shows the statistics of the ontologies generated from the three data sets. The hierarchy of the ontology generated from the PROGRAMMING LANGUAGE data sets has 6 levels from the root concept to the bottom level concepts and contains 422 concepts (except bottom level concepts). The hierarchy of the ontology generated from the SPORT data sets has 5 levels and contains 273 concepts (except bottom level concepts). The hierarchy of the ontology generated from the GAME data sets has 5 levels and contains 313 concepts (except bottom level concepts).

Using ODP as the gold standard for evaluation, we apply F1 score [13] to compare the ontology built by our approach with ODP. F1 score is a measure of a categorization result's accuracy according to the standard. It is the harmonic mean of precision and recall. If the ontology is more similar to ODP, the F1 score will be higher, which means the ontology is more consistent with human thinking. In previous research of ontology learning from folksonomies [14][15], researchers ignore the instances and categories. They define tags as concepts and only explore the relationship between these tags. There is not any category structure in the ontology generated by previous approaches. Their methods cannot organize instances into a category structure as ours. Accordingly it is impossible to compare the category structure of the ontology generated by our method with them. As commonly used clustering methods, K-means and concept clustering algorithm COBWEB can cluster instances into different categories. We compare the category structure built by our method with that built by K-means and COBWEB to demonstrate the effectiveness of our approach on categorization.

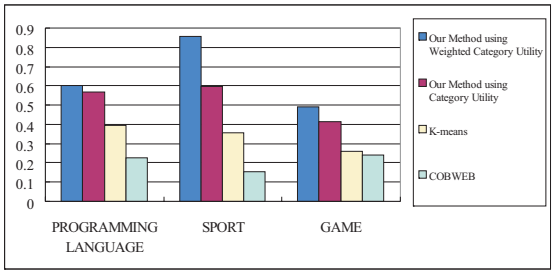

Figure 2: Comparison on F1-scores.

In Figure 2, we show F1 scores of the results using different algorithms in the three data sets (PROGRAMMING LANGUAGE, SPORT and GAME). It is observed that our algorithm performs better than others especially in the SPORT data set (0.855) which means the category structure built by our method is more similar to ODP than others. In sports domain, the basic level categories are explicit so that they can easily be detected. Basketball, football, running and other types of sports form the basic level categories in this domain (referring to table 2). In addition, the content of web pages in sports domain is unambiguous and the noise tags are fewer than in other domains. The result in the GAME data set is not as good as others because the ODP categories in this domain do not lay on the basic levels in our opinion. The F1 score of the results using our approach in the PROGRAMMING data set is 0.604 which

Table 2: Similar Concept Pairs between ODP and Ontologies generated by Our Method

	ODP	Ontologies generated by Our Method
	sub-concepts of (programming)	sub-concepts of (programming:0.3)
1	(c-sharp)	(.net:0.349)
2	(assembly)	(assembly:0.508, asm:0.244, assembler:0.256, development:0.105)
3	(c++)	(c++:0.641, development:0.155)
4	(c)	(c:0.522)
5	(pl-sql)	(database:0.450, development:0.100)
6	(sql)	(erlang:0.889)
7	(java)	(java:0.730)
8	(javascript)	(javascript:0.704)
9	(lisp)	(lisp:0.661)
10	(perl)	(perl:0.800)
11	(php)	(php:0.745)
12	(python)	(python:0.853)
13	(ruby)	(ruby:0.690)
14	(scripting)	(scripting:0.280)
15	(delphi)	(software:0.173, development:0.178, delphi:0.743)
	sub-concepts of (sports)	sub-concepts of (sport:0.498)
1	(Baseball)	(baseball:0.736)
2	(Basketball)	(basketball:0.535)
3	(Boxing)	(boxing:0.695)
4	(Cricket)	(cricket:0.698)
5	(Cycling)	(cycling:0.425, bike:0.395)
6	(football)	(soccer:0.397, football:0.459)
7	(golf)	(golf:0.809)
8	(hockey)	(hockey:0.603)
9	(Martial_Arts)	(martialart:0.299, martial_art:0.136)
10	(Motorsports)	(racing:0.325, new:0.215, motorsport:0.266)
11	(running)	(running:0.708, fitness:0.229)
12	(Water_Sports)	(surf:0.448, surfing:0.454)
	sub-concepts of (games)	sub-concepts of (game:0.417)
1	(online)	(free:0.184, online:0.065)
2	(gambling)	(gambling:0.337)
3	(card_games)	(poker:0.883)
4	(roleplaying)	(rpg:0.442)
5	(puzzles)	(puzzle:0.421)
6	(board_games)	(chess:0.802)
	...	...

is about 50% higher than the results using K-means. We also compare our approach with COBWEB [16] which is an incremental conceptual clustering algorithm also aiming to maximize category utility as our approach. In COBWEB, they use a incremental strategy to add instances to the category structure one by one. Although this strategy is flexible, the limitation is that the structure determined in previous steps is fixed. Accordingly, the order of the instances will impact the quality of the result which makes the quality uncertain. To solve this problem and improve the quality, our approach considers the whole data set first and always merges the most similar ones together. This strategy makes sure that we are finding the basic level categories in the whole data set. In addition, our method performs better using weighted category utility as the metric than using category utility in the three data sets. This is because weighted category utility considers the difference of tags, which is the situation in folksonomies.

4.3. Qualitative Analysis

In this section, we will discuss the quality of the ontologies generated by our method. The ontology generated by our method is similar to ODP ontology. Table 2 shows the similar pairs between ODP concepts and concepts in the ontologies generated by our method. Concepts generated by our method are described in the form *(tag:value,...,)*. Concepts in ODP are described in the form *(label)*. The tags from super-concepts are not shown in the table because of the limit of space, e.g. the concept *(.net:0.349)* should be *(programming:0.415, .net:0.349)*. Most sub-concepts of *(programming:0.3)* are about programming languages in this data set, such as Java, Python and Ruby. This is consistent with the basic level concepts of programming language domain in human thinking. As shown in table 2, properties of these concepts are related with labels of ODP concepts. There are totally 15 similar pairs (47% of the sub-concepts). In addition, in the SPORT data set, there are 12 similar pairs (23% of the sub-concepts) and in the

GAME data set, there are 6 similar pairs (37.5% of the sub-concepts). These similar concepts and the relations of concepts demonstrate that our method is effective on generating ontologies with basic level concepts and the generated ontologies are meaningful and consistent with human thinking.

According to the research of Zhou et al. [15], we notice that the relations between different tags or concepts mainly include three types. (1) B is the sub-type of A. (e.g. "java" is sub-type of "programming") (2) B is a related aspect of A. (e.g."development" is related with "programming") (3) B is parallel to A. (e.g. "java" is parallel to "python"). According to the definition of ontologies, the relations between concepts of different levels should be type 1. To demonstrate the effectiveness of our approach on generating hierarchical structure of ontologies, we compare the relations between first level concepts and second level concepts in the ontology generated by our method with that generated by Zhou's method. The result is shown in Figure 3. The result shows that the percentage of type 1 (sub-type) relation in the ontology generated by our method (79%) is much higher than that generated by Zhou's method (30%). The percentage of type 2 relation is 21% and 70% respectively. In addition in this situation, there is no type 3 relation. The result demonstrates that the hierarchical structure in the ontology generated by our method, to some extent, is better than that generated by Zhou's method.

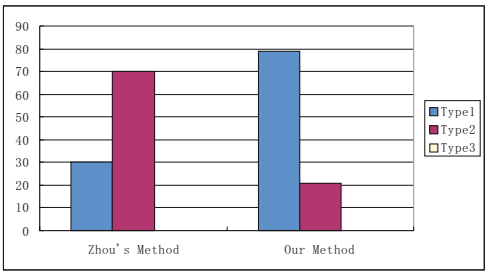

Figure 3: Comparison of Relations.

5. Application

Ontologies generated by our approach have many immediate applications, such as collaborative tagging, tag aided search and tag recommendation. The semantic relationship among tags defined in these ontologies can specify the searching and crawling process. As an example, if a search engine is asked to find some web pages about programming languages, according to the ontologies generated in the PROGRAMMING LANGUAGE data set, the engine will notice that the sub-concept of "programming language" such as "Java", "C" and "PHP" are related with its target. These ontologies can also be used for knowledge representation in Semantic Web, B2B interaction among sites and multi-agents communication. In addition, our approach can be used to categorize different resources such as photos, books and movies in Internet.

6. Related Work

Much research has been conducted on semantics in folksonomies. An approach for making explicit the semantics behind the tag space through mapping folksonomies to existent ontologies are presented by Specia et al. [17]. Jaschke et al. [18] defined a new data mining task, the mining of frequent tri-concepts, and presented an efficient algorithm to discover these implicit shared conceptualizations. Furthermore, there are many research works focusing on ontology learning from folksonomies. Damme et al [19] sketch a comprehensive approach for deriving ontologies from folksonomies by integrating multiple resources and techniques. Mika [14] use set theory to extract broader/narrower tag relations and propose an approach to extend the traditional bipartite model of ontologies with the social annotations. Zhou et al. [15] applied Deterministic Annealing for clustering tags and building tags hierarchical structure in a top-down method.

7. Conclusion

This paper presents a novel idea to make use of implicit semantics in folksonomies to build ontologies. Inspired by studies in cognitive psychology, we present an algorithm to generate ontologies with basic level concepts. This type of ontology is considered as cognitive basic and more acceptable and applicable by users. Furthermore, we generated ontologies based on folksonomy tags which agree more closely with human thinking than those automatically extracted from text. To the best of our knowledge, it is the first work on discovering basic level concepts in folksonomies and using them to construct ontologies. In experiments, ontologies generated from three real-world data sets demonstrate the effectiveness of our approach on generating ontologies with basic level concepts.

8. Acknowledgements

This work presented in this paper is partially supported by a CUHK Direct Grant for Research, and it has been supported, in part, by a research grant from Hong Kong Research Grants Council (RGC) under grant CityU 117608.

References

[1] H. Al-Khalifa, H. Davis, Exploring the value of folksonomies for creating semantic metadata, International Journal on Semantic Web & Information Systems 3 (1) (2007) 12–38.
[2] P. Buitelaar, P. Cimiano, B. Magnini, Ontology learning from text: An overview, Ontology learning from text: Methods, evaluation and applications (2005) 3–12.
[3] E. Rosch, C. B. Mervis, W. D. Gray, D. M. Johnson, P. Boyes-Braem, Basic objects in natural categories, Cognitive Psychology (3) 382–439.
[4] G. Murphy, The big book of concepts, Bradford Book, 2004.
[5] C. Cattuto, D. Benz, A. Hotho, G. Stumme, Semantic grounding of tag relatedness in social bookmarking systems, in: Proceedings of ISWC, Springer, 2008.
[6] T. Lee, J. Hendler, O. Lassila, et al., The semantic web, Scientific American 284 (5) (2001) 34–43.
[7] G. Antoniou, F. Van Harmelen, A semantic web primer, MIT press, 2004.
[8] Y. Cai, H. F. Leung, A formal model of fuzzy ontology with property hierarchy and object membership, in: ER '08: Proceedings of the 27th International Conference on Conceptual Modeling, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 69–82.
[9] C. M. Au Yeung, H. F. Leung, Ontology with likeliness and typicality of objects in concepts, Lecture Notes in Computer Science 4215 (2006) 98.
[10] S. Golder, B. Huberman, The structure of collaborative tagging systems, Arxiv preprint cs/0508082.
[11] M. Gluck, J. Corter, Information, uncertainty, and the utility of categories, in: Proceedings of the seventh annual conference of the cognitive science society, 1985, pp. 283–287.
[12] J. Schultz, M. Liberman, Topic detection and tracking using idf-weighted cosine coefficient, in: Broadcast News Workshop'99 Proceedings, 1999, p. 189.
[13] C. Manning, P. Raghavan, H. Schtze, Introduction to information retrieval, Cambridge University Press, 2008.
[14] P. Mika, Ontologies are us: A unified model of social networks and semantics, Web Semantics: Science, Services and Agents on the World Wide Web 5 (1) (2007) 5–15.
[15] M. Zhou, S. Bao, X. Wu, Y. Yu, An unsupervised model for exploring hierarchical semantics from social annotations, Lecture Notes in Computer Science 4825 (2007) 680.
[16] D. Fisher, Knowledge acquisition via incremental conceptual clustering, Machine learning 2 (2) (1987) 139–172.
[17] L. Specia, E. Motta, Integrating folksonomies with the semantic web, Lecture Notes in Computer Science 4519 (2007) 624–639.
[18] R. Jäschke, A. Hotho, C. Schmitz, B. Ganter, G. Stumme, Discovering shared conceptualizations in folksonomies, Web Semantics: Science, Services and Agents on the World Wide Web 6 (1) (2008) 38–53.
[19] C. Van Damme, M. Hepp, K. Siorpaes, Folksontology: An integrated approach for turning folksonomies into ontologies, Bridging the Gap between Semantic Web and Web 2 (2007) 57–70.