An Unsupervised Model for Exploring Hierarchical Semantics from Social Annotations

Mianwei Zhou, Shenghua Bao, Xian Wu[**], and Yong Yu

APEX Data and Knowledge Management Lab
Department of Computer Science and Engineering
Shanghai Jiao Tong University, 200240, Shanghai, P.R. China
{kopopt,shhbao,yyu}@apex.sjtu.edu.cn wuxian@cn.ibm.com

Abstract. This paper deals with the problem of exploring hierarchical semantics from social annotations. Recently, social annotation services have become more and more popular in Semantic Web. It allows users to arbitrarily annotate web resources, thus, largely lowers the barrier to cooperation. Furthermore, through providing abundant meta-data resources, social annotation might become a key to the development of Semantic Web. However, on the other hand, social annotation has its own apparent limitations, for instance, 1) ambiguity and synonym phenomena and 2) lack of hierarchical information. In this paper, we propose an unsupervised model to automatically derive hierarchical semantics from social annotations. Using a social bookmark service Del.icio.us as example, we demonstrate that the derived hierarchical semantics has the ability to compensate those shortcomings. We further apply our model on another data set from Flickr to testify our model's applicability on different environments. The experimental results demonstrate our model's efficiency.

1 Introduction

Social annotation services have recently attracted considerable users and interest. Prominent web sites like Flickr[1], Del.icio.us[2] are widely used and achieve significant success. These services not only provide user-friendly interfaces for people to annotate and categorize web resources, but also enable them to share the annotations and categories on the web, encouraging them to collaboratively enrich meta-data resources. In 2004, Thomas Vander Wal named these services "Folksonomy", which came from the terms "folk" and "taxonomy" [1].

Compared with the traditional meta-data organization, folksonomy represents high improvement in lowering barriers to cooperation. Traditional taxonomy, which is predefined only by small groups of experts, is limited and might easily become outdated. Social annotation just solves these problems by transferring the burden from several individuals to all web users. Users could arbitrarily

---

[**] Xian Wu is now working in IBM China Research Lab.

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

[2] http://del.icio.us

annotate web resources according to their own vocabularies, and largely enrich the meta-data resources for Semantic Web.

However, although social annotation services have large potential to boom the Semantic Web, development of these services are impeded by their own shortcomings. Such shortcomings are mainly due to two features of folksonomy:

- Uncontrolled vocabulary. Breaking away from the authoritatively determined vocabulary, folksonomy suffers several limitations. One is ambiguity. People might use the same word to express different meanings. Another phenomenon is synonym. Different tags might denote the same meaning. With ambiguity and synonym, users might easily miss valuable information while gain some redundant information.
- Non-hierarchical structure. Folksonomy represents a flat but not hierarchical annotation space. This property brings difficulties in browsing those systems, moreover, makes it hard to bridge folksonomy and traditional hierarchical ontologies.

Aimed at overcoming those shortcomings, many researches have been conducted, for instance [2,3,4]. [2] introduced the concept of "navigation map" which described the relationship between data elements. The author showed how to gain semantic related images when users made queries. [3] gave a probabilistic method to allocate tags into a set of parallel clusters, and applied these clusters to search and discover the Del.icio.us bookmarks. Both of [2] and [3] focused on exploring relations between tags in the uncontrolled vocabulary, but still did not solve the non-hierarchy problem. In [4], the author proposed an algorithm to derive synonymic and hierarchical relations between tags, and demonstrated promising results. But the model is supervised, thus could not be effectively extended to other contexts, and also, lacks a sound theoretical foundation.

In our paper, we propose an unsupervised model, which could automatically derive hierarchical semantics from the flat tag space. Although search engines which aim to derive hierarchy out of search results have already existed(e.g. Vivísimo[3]), to the best of our knowledge, no work has been done before on exploring hierarchical semantics from tags. We demonstrate that the derived hierarchical semantics well compensates folksonomy's shortcomings.

In order to derive the hierarchical semantics, our model proceeds in a top-down way. Beginning with the root node containing all annotations, we apply the splitting process to gain a series of clusters, each of which represents a specific topic. Further apply the splitting process on each cluster, smaller clusters with narrower semantics are gained. It's easy to observe, this recursive process helps us obtain a hierarchal structure. A probabilistic unsupervised method named Deterministic Annealing(DA) algorithm is utilized in each splitting process. Unlike other clustering algorithm, DA algorithm could well control the cluster number and each cluster's size with the help of a parameter T. We make use of this feature to ensure that each node's semantics could be identified by a few tags.

Different from previous work, our model has several important features:

---

[3] http://vivisimo.com/

- Unsupervised model. Without any need of training data, it could be easily extended to other social annotation services.
- Hierarchical model. In the derived structure, each node represents an emergent concept and each edge denotes the hierarchical relationship between concepts.
- Self-controlled model. In our model, the number and the size of clusters are automatically determined during the annealing process.

The hierarchical semantics derived from our model has a large number of applications. Take two for example: 1)Semantic Web. The derived hierarchical semantics well serves as a bridge between the traditional strict ontology and the distributed social annotations. It would make ontology more sensitive to users' interests and demands, and reflect the current trends in the Internet; 2)Resource Browsing & Organization. The derived hierarchical semantics could also be utilized as effective tools for resources browsing and organization. Users could easily trace the path from the root to the node which contains information they want.

The rest of the paper is organized as follows. Section 2 briefly reviews the previous study of social annotation and DA algorithm. Section 3 gives a detailed description of our algorithm. Section 4 gives the experimental results and related evaluations. Finally we make a conclusion in Section 5.

2   Related Work

2.1   Related Work on Social Annotation

In these years, social annotation becomes a hot topic, on which many researches have been conducted. Part of these researches focused on discussing features of social annotations. [5,6] pointed out the advantages and limitations of social annotation, and described the contribution it would make to World Wide Web. [7] gave a brief review of those social annotation services available on network. In [8], the author discovered statistical regularities behind those collaborative tagging systems, and predicted the stable patterns through a dynamic model. [9] improved [8]'s work. The author showed the regularity behind those services could be described by a power law distribution. Furthermore, it showed that co-occurrence networks could be utilized to explore tags' semantic meaning.

For Semantic Web, the metadata resources usually exist as a form of predefined ontology. As social annotation services popularize, researchers aim to derive emergent semantics[10] from those systems, and utilize the derived structure to enrich the Semantic Web(e.g. [11,3,12]). [11] proposed an approach to extend the traditional bipartite model of ontologies with the social annotations. [3,12] are similar with our work. They respectively proposed model to derive emergent semantics from social annotations. However, in [3], the derived structure was still flat but not hierarchical. In [12], although the author constructed a topical hierarchy among tags, the derived structure was a simple binary tree, which might not be applicable for some complex social annotation environments.

Different from their work, we propose a novel model to derive hierarchical semantics which could effectively reflect the semantic concepts and hierarchical relationship from social annotations.

In addition to Semantic Web, some researches aimed at facilitating the social annotation application itself. In [2], the author proposed a similarity search model that allowed users to get concept-related data elements. [4] further explored the hierarchical relation between tags. [13] changed the perspective. The author presented a model to visualize the evolution of tags on the Flickr, thus users could gain the hottest images in any time interval. In [14], the author presented a model named FolkRank to exploit the structure of the folksonomy. [15] proposed two algorithms to incorporate the information derived from social annotations into page ranking.

2.2   Deterministic Annealing Background

The key algorithm in our model is named Deterministic Annealing (DA). It is an algorithm motivated by physical chemistry and mainly based on the information theory. In computer science, DA was widely utilized in the area of computer linguistics , computer vision and machine learning (e.g.[16,17,18,19]).

3   The Proposed Method

In this section, we give a detailed description of our model. The social annotations we use as our data set come from a popular bookmark service called Del.icio.us. It is very easy to extend our model to other common social annotation services such as Flickr, Technorati and so on.

3.1   Data Analysis

Del.icio.us is a social bookmark web service for sharing web bookmarks. Users could not only store and manage their own bookmarks, but also access others' bookmark storage at any time[20]. It is a flexible and useful tool for users with similar interests to share topics.

The data in Del.icio.us could be described as a set of quadruples:

$$(user, tag, website, time)$$

which means that the *website* is annotated by the *user* with the *tag* at the specific *time*. In our model, we focus on the tag and web elements. Let us denote the set

$$S_{tag} = \{t_1, t_2, ..., t_N\}, S_{website} = \{w_1, w_2, ..., w_M\}$$

$$S_{pair} = \{\langle t(i), w(i)\rangle | i \in [1, L]\}$$

where N, M, L respectively represent the number of tags, websites and pairs, and $\langle t(i), w(i)\rangle$ represent that the ith pair includes the $t(i)$th tag and the $w(i)$th website.

3.2   Algorithm Overview

Our model builds the hierarchical structure in a top-down way. Beginning with the root node, the model recursively applies splitting process to each node until termination conditions are satisfied. In each splitting process, Deterministic Annealing(DA) algorithm is utilized. Figure 1 gives an intuitive description of this splitting process.
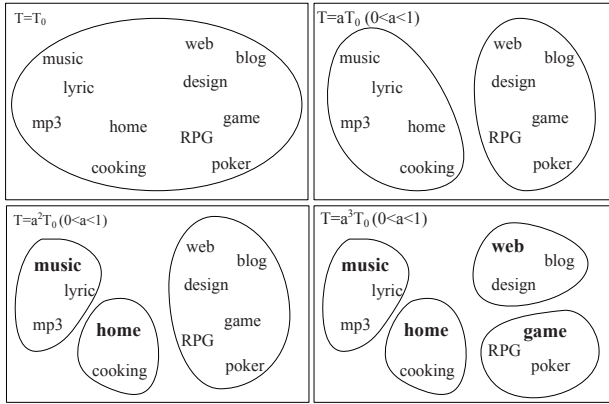

Fig. 1. The Emergent Semantics during the Annealing Process

In Figure 1, we observe that controlled by a parameter T, DA algorithm splits the node in a gradual way. As T is lowered from the first to the fourth subgraph, the cluster number increases from one to four finally. This process terminates when all clusters become "Effective Cluster", or the number of "Effective Cluster" reaches a upper bound. The term "Effective Cluster" refers to those clusters whose semantics could be generalized by some specific tags. We name those tags "Leading Tag" for this cluster. It should be noted that the effective clusters do not emerge immediately. In the second sub-graph, neither of the clusters are effective clusters, because their semantics are too wide to be generalized by any tag. In the fourth sub-graph, all clusters are effective clusters, leading tags for which are "music", "home", "web" and "game" respectively. In our model, we design a criterion, which is given in section 3.4, to identify an effective cluster.

An overview of our model is given in Algorithm 1. In Algorithm 1, we maintain a queue Q to store the information of nodes which are waiting for splitting. Vector $\boldsymbol{P}$ in the queue indicates the probability that each tag emerges in this node. At line 1, elements of $\boldsymbol{P}_0$ are all initialized with 1, because all tags are contained in the root node. From line 2 to 10, the algorithm recursively splits each node until the termination condition is satisfied. We finally gain a hierarchical structure and each node's semantics is identified by its corresponding leading tags.

Line 4 is a key part of our model. The function f_D serves as a clustering machine. Input the node's information, and f_D outputs a series of effective clusters

Algorithm 1. Deriving Hierarchical Semantics

1: Initialize Q. Q is a queue containing one N dimensions vector $\boldsymbol{P}_0 = (1, 1, ..., 1)$
2: **while** Q is not empty **do**
3:     Pop $\boldsymbol{P}$ from Q. Let $\boldsymbol{P} = (p_0, p_1, ..., p_N)$.
4:     $\{p(c_i|t_j)|i \in [1, C], j \in [1, N]\} \leftarrow f_D(\boldsymbol{P})$
5:     **for** each cluster c_i, $i = 1, 2, ..., C$ **do**
6:         Extract leading tags t_{c_i} to stand for the semantics of cluster c_i
7:         **if** c_i could be further split **then**
8:             Let $\boldsymbol{P}' = (p'_0, p'_1, ..., p'_N)$

$$p'_j = \begin{cases} p_j * p(c_i|t_j) & t_j \neq t_{c_i} \\ 0 & t_j = t_{c_i} \end{cases}$$

            Push $\boldsymbol{P}'$ into Q.
9:         **else**
10:           The remaining tags except leading tags t_{c_i} form leaves for the current node.
11:         **end if**
12:     **end for**
13: **end while**

---

derived from this node. Each cluster is described by the value $p(c_i|t_j)$ represents the relativity between the jth tag and the ith cluster. As discussed before, DA algorithm is utilized in f_D. Detailed implementation of this algorithm is given in the following section. The termination condition for DA algorithm is given in section 3.4.

3.3 Apply Deterministic Annealing for Clustering

In this section, we introduce how to apply Deterministic Annealing(DA) algorithm to split a tag set on a node into several effective clusters. In mathematics, DA and other similar optimizing algorithms could all be stated as a process to minimize a predefined criterion. In our model, such criterion is given below:

$$D = \sum_{i=1}^{N} \sum_{j=1}^{C} p(c_j|t_i) * d(t_i, c_j) \tag{1}$$

where $d(t_i, c_j)$ measures the relativity between tag t_i and the cluster c_j. We used KL-divergence to describe this distance.

$$d(t_i, c_j) = \sum_{k=1}^{M} p(w_k|t_i) * log(\frac{p(w_k|t_i)}{p(w_k|c_j)}) \tag{2}$$

where $p(w|t_i)$ and $p(w|c_j)$ respectively measure tag t_i's and cluster c_j's distributions on all websites. Through measuring KL-divergence between these two distributions, we gain the semantic distance between tag t_i and the concept that cluster c_j represents. With closer semantic relation between them, $d(t_i, c_j)$

becomes smaller. It is easy to observe, as D is minimized, the value of $p(c|t)$ indicates a clustering result.

In the minimizing process of D above, general clustering algorithm might easily suffer a poor local minimum. In order to overcome this problem, DA recasts the minimization problem by introducing an annealing process. The minimization of D is converted to the minimization of free energy F subject to a specified level of randomness.

$$F = D - TH \tag{3}$$

H is a measure of level of randomness, given below

$$H = -\sum_{i=1}^{N}\sum_{j=1}^{C} p(c_j|t_i) * log[p(c_j|t_i)] \tag{4}$$

Free energy F and entropy H are two terms in the physical annealing theory. Temperature T could control entropy H in different scales during the minimization of F. As T is lowered, H also decreases. As illustrated in Figure 1, with low entropy H, every tag is more definitely linked to clusters, resulting in the increment of cluster number.

Algorithm 2. Apply DA Algorithm for Clustering

1: Input: $\boldsymbol{P}$
2: $C \leftarrow 2$, $T \leftarrow T_0$.
3: Set $p(c_i|t_j)$ with random values between 0 and 1, satisfying $\sum_{i=1}^{C} p(c_i|t_j) = 1$, for all $j = 1, 2, ..., N$.
4: **loop**
5:     $p^{(0)}(c_i|t_j) \leftarrow p(c_i|t_j)$, $k \leftarrow 0$, calculate $F^{(0)}$.
6:     **repeat**
7:         $k \leftarrow k + 1$
8:         Calculate $p^{(k)}(c_i|t_j)$ with $p^{(k-1)}(c_i|t_j)$ according to Equation (6)
9:         Calculate $F^{(k)}$ according to Equation (3)
10:     **until** $|F^{(k)} - F^{(k-1)}| < \epsilon$
11:     Let $p^{(K)}(c|t)$ be the final iteration result.
12:     **if** all clusters are effective clusters **then**
13:         **return** $p^{(K)}(c|t)$
14:     **end if**
15:     **if** Critical Temperature for cluster c_i is reached **then**
16:         $p(c_{C+1}|t_j) \leftarrow p(c_i|t_j)/2 + \delta$, $p(c_i|t_j) \leftarrow p(c_i|t_j)/2 - \delta$, where δ indicates a random perturbation.
17:         $C \leftarrow C + 1$
18:     **else**
19:         $p(c_i|t_j) \leftarrow p^{(K)}(c_i|t_j)$
20:     **end if**
21:     $T \leftarrow \alpha T (0 < \alpha < 1)$
22: **end loop**

A detailed implementation of DA is given in Algorithm 2 as a supplement of line 4 in Algorithm 1. In Algorithm 2, line 1 is the input $\boldsymbol{P}$ which contains the information of the node waiting for splitting. Line 2 to 3 are the initialization steps. Line 4 to 21 represent the annealing process of the algorithm. Among them, Expectation-Maximum(EM) algorithm is utilized to minimize the free energy F in line 5 to 11. The termination condition for this algorithm is given in line 12 to 14. From line 15 to 20, we determine when the cluster number should be increased. In line 19, temperature T is lowered preparing for next annealing process. In the following section, we would further discuss the detail about minimizing F and determining the increment of the cluster number.

EM Algorithm for Minimizing F. We utilize EM algorithm to iteratively minimize F. Firstly, the equation (3) is recast as

$$F = \sum_{i=1}^{N}\sum_{j=1}^{C} p(c_j|t_i) * (\sum_{l=1}^{M} p(w_l|t_i) * log(\frac{p(w_l|t_i)}{p(w_l|c_j)}) + T * log(p(c_j|t_i))) \quad (5)$$

Through EM algorithm, $p(c|t)$ could be estimated by iteratively minimizing the free energy F. Beginning with the initial value for $p^{(0)}(c_i|t_j)$, we give the $p^{(k)}(c_i|t_j)$ in the kth iteration.

$$p^{(k)}(c_i|t_j) = \frac{exp(-\frac{d^{(k)}(t_j,c_i)}{T}) * p^{(k)}(c_i)}{\sum_{l=1}^{C} exp(-\frac{d^{(k)}(t_j,c_l)}{T}) * p^{(k)}(c_l)} \quad (6)$$

where

$$p^{(k)}(c_i) = \sum_{j=1}^{N} p^{(k-1)}(c_i|t_j) * p(t_j) * p_j \quad (7)$$

$$p^{(k)}(w_l|c_i) = \frac{\sum_{j=1}^{N} p^{(k-1)}(c_i|t_j) * p(t_j) * p_j * p(w_l|t_j)}{p^{(k)}(c_i)} \quad (8)$$

$$d^{(k)}(t_j, c_i) = \sum_{l=1}^{M} p(w_l|t_j) * log(\frac{p(w_l|t_j)}{p^{(k)}(w_l|c_i)}) \quad (9)$$

where, $p(c)$ denotes the probability that the cluster is assigned. $p(t)$ denotes the probability that the tag occurs in the data set. p_i denotes the probability that the ith tag occurs in the current sub-node. $p(w|t)$ denotes the relativity between the website and the tag. Among them, $p(w|t)$ and $p(t)$ are invariants which could be computed directly from the data set, while $p(c)$, $p(w|c)$, and $d(t,c)$ are variant, which are converging during the whole iteration process. Given P and T, F finally converges to a minimum after a series of iterations. For further details about the derivation of the formulas, refer to [19].

Critical Temperature Determination. From line 15 to 20 in Algorithm 2, we introduce a new concept "Critical Temperature". In the DA algorithm theory, once the temperature reaches certain clusters' critical temperature, those clusters should be split, so that the Free Energy could be further minimized. This process is named "Phase Transition". The increment of cluster number in DA algorithm is achieved by a series of phase transitions. It has been theoretically proved that this critical temperature could be calculated, but the computation is too complex. [19] introduced a simple alternative to estimate critical temperature. In this method, an extra copy is kept for each cluster. Only when the critical temperature is reached for a cluster, its copy would split away, otherwise, the copy would merge again after the iteration. We utilize this method in our model. Once phase transition for certain clusters is detected, we add a new cluster in line 16.

3.4 Effective Cluster Identification

As discussed in the previous section, DA algorithm in our model terminates only when all clusters become effective clusters, or the number of effective clusters reaches an upper bound. In this section, we give a criterion to identify whether a cluster is effective.

The main difference of the effective cluster from other common ones is that, as an effective cluster, its semantics could be generalized by some specific tags, which we name "Leading Tag". To measure a tag's capability to summarize the whole cluster's semantics, we define $Cov(t_i, c_j)$ to measure a tag's coverage as below.

$$Cov(t_i, c_j) = \sum_{k=1}^{N} p(t_k|c_j) b_{i,k} \tag{10}$$

where, $b_{i,k} \in \{0, 1\}$ indicates whether there exists a website annotated by both tags t_i and t_k. $p(t|c)$ could be easily gained by applying Bayesian Theorem on $p(c|t)$. The high value of $Cov(t_i, c_j)$ indicates that tag t_i has covered lots of other tags in cluster c_j, so t_i is more capable to summarize cluster c_j's semantics. Using $Cov(t_i, c_j)$, $E(c_j)$ measuring whether c_j is an effective cluster is defined.

$$E(c_j) = \max_{i \in [1,N]} Cov(t_i, c_j) \tag{11}$$

The qualification for a cluster to be an effective one is measured by the leading tag with highest $Cov(t_i, c_j)$ in it. If multiple leading tags are allowed, $E(c_j)$ could also be measured by several largest ones. During the annealing process, $E(c_j)$ increases as the size of clusters is reduced. Once $E(c_j)$ reaches a high value, it indicates that the leading tag t_{c_j} has emerged, so we accept this cluster as an effective cluster.

4 Experiment

4.1 Experiment Setup

Our experiment is mainly conducted on two samples of Data: Del.icio.us and Flickr. We filter those tags and urls which emerge less than 20 times in the data set. The statistics for both of the raw and the filtered data is present in Table 1.

Table 1. Statistics of Data Sets

Source	Raw Data			Filtered Data			Crawled Time
	tag	url	pair	tag	url	pair	
Del.icio.us	192143	784617	3357809	8445	16963	479035	April 2006
Flickr	32465	23713	204717	3927	6127	70761	April 2007

4.2   Experiment on Del.icio.us

Derived Hierarchical Structure. We apply our model on the Del.icio.us data set described above. Figure 2 shows part of the derived hierarchical result.
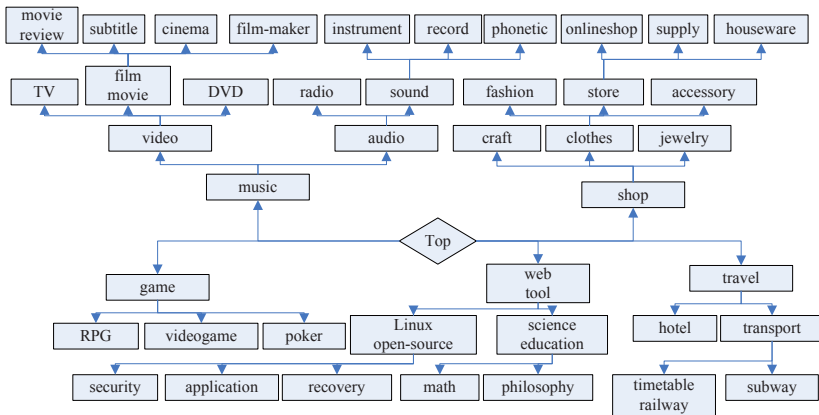
Fig. 2. Hierarchical Semantics Derived from Del.icio.us

In Table 2, we randomly choose some nodes from each hierarchy, and display their locations and child-clusters. Each node "(tag1, tag2,...)" in Table 2 denotes a cluster with several leading tags. In Figure 2 and Table 2, we observe that the derived hierarchical semantics is well matched with people's common knowledge.

Because our model is based on statistics about human behaviors, it is hard to restrict the derived relationship to a specific type. In further experiment, we discover that the hierarchical relationship mainly includes three types. Suppose B is the child node of A

1. B is the sub-type of A(e.g. "RPG" and "videogame" are both "game").
2. B is the related aspect of A(e.g. "hotel" and "transportation" to "travel").
3. B is parallel to A(e.g. the sub-node of "DVD" is "WMA", "DV").

In Figure 3, we present a statistics of each type's portion between different hierarchies. It's observed that type 1 and 2 mainly exist in the higher level of the tree, and type 3 exists in the lower level. Although type 3 deviates our original purpose, we should not expect our model to derive a precise ontology like Wordnet containing only type 1 and 2. When the semantics of a node becomes

Table 2. Clusters in Different Hierarchies

| Leading Tag	Ancestor Node	Child Node
food, health	⟨Top⟩	(fit), (sport), (eat, bread, coffee), (cook, recipe), (beer)
politics	⟨Top⟩	(government), (law, right), (active), (censorship), (conspiracy, 911), (Israel, Iran, Syria), (military, war), (Africa), (habitat, human)
language	⟨Top⟩ → (web, tool)	(write), (English, linguist, word), (translate), (encyclopedia), (Chinese, Mandarin)
jewelery	⟨Top⟩ → (shop)	(Chicago, glass), (ear, bracelet, bridal, necklace, ring), (handmade), (unusual), (stainless, diamond)
webdesign, webdevise	⟨Top⟩ → (web, tool) → (program, develop)	(html, xhtml, standard), (ajax, xml), (tutorial, code, opensource), (sql, mysql), (framework, python), (menu, navigate), (color, palette), (encode, unicode, UTF8)
DVD	⟨Top⟩ → (music) → (video)	(WMA, MP4, quicktime), (DV, camcorder, miniDV), (codec, Divx, mpeg, avi)
cryptography, encrypt	⟨Top⟩ → (web, tool) → (Linux, opensource) → (security)	(PKI), (computers and internet), (GPG, GNUPG), (MD5), (OpenSSL)
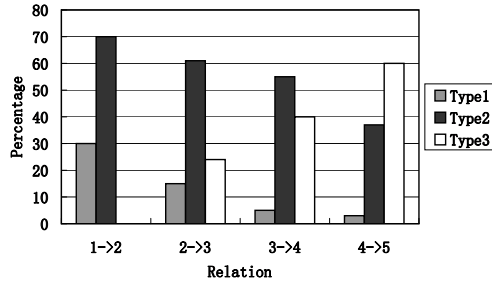
Fig. 3. Statistics for Each Type of Relation between Different Hierarchy Levels

narrower in lower level, it is a hard task to select leading tags to summarize the semantics of the node by human, let alone by computer.

Distribution of Tags on Different Nodes. The distribution of tags on different nodes is also studied. In Table 3, we randomly select some tags and give their linked clusters with largest probabilities. For those well-known polysemantic words (e.g. "wine", "apple"), their diverse meanings could be observed through different paths. For other common words, different nodes could represent their distinct related aspects. For instance, the word "honeymoon" is related not only to "travel" and "holiday", but also to "gift". This feature of our model well solves the ambiguity problem. In the derived hierarchical structure, a lot of tags has more than one related node, but at most five. It is because when temperature is lowered in the iterative steps, tags would easily converge to one or two clusters, but not scatter equally on several ones.

Table 3. Distribution of Tags on Different Nodes

Tags	Distribution on Different Nodes
agriculture	1. (environment) → (sustain, green) →(agriculture)
	2. (food) → (garden) → (agriculture, farm)
wine	1. (web, tool) → (Linux, opensource) → (freeware)→(Wine)
	2. (food) → (coffee, eat, tea)→(wine)
price	1. (money, finance) → (bill) → (price)
	2. (shop) → (deal, buy) →(price)
gasoline	1. (shop) → (deal, buy) → (gasoline)
	2. (travel) → (transport) → (automobile) → (gasoline)
honeymoon	1. (travel) → (hotel) → (holiday) → (honeymoon)
	2. (gift)→ (jewelry) → (bridal, wed) → (honeymoon)
apple	1. (web, tool) → (Linux, open-source) → (Apple, Mac)
	2. (food) → (coffee, eat, tea) → (apple)

4.3   Experiment on Flickr

We also apply our model on a sample of Flickr data set to demonstrate our model's wide applicability. With effective self-controlled capability, our model well captures different features of social annotation environment in Flickr. Figure 4 gives part of the result.
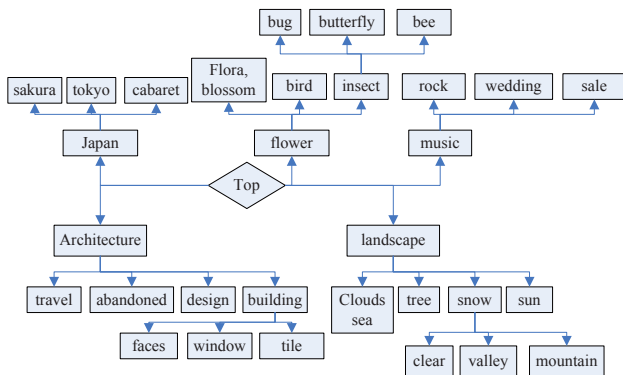
Fig. 4. Hierarchical Semantics Derived from Flickr

Form Figure 4, we discover that the derived relation is reasonable according to people's knowledge. Compared with the structure derived from Del.icio.us, the number of derived hierarchical relations is much less. Most of the nodes concentrate on the first and second hierarchies with parallel relations. It is mainly because Flickr is a "Narrow Folksonomy"[21] compared with the "Broad Folksonomy" Del.icio.us. In the Narrow Folksonomy, most of the tags are singular

and directly linked to the object. This property largely limits the hidden semantics in social annotations. However, our model still captures the hidden topics behind Flickr and presents a satisfying hierarchical result.

5   Conclusion and Future Work

Social annotation has become more and more popular because of its strengths. But at the same time, it also has its own shortcomings, for instance, 1)ambiguity and synonymous phenomena 2)non-hierarchical structure. In order to overcome these shortcomings, we build an unsupervised model to derive hierarchical semantics from social annotations. The main contributions can be concluded as follows:

1. The proposal to study the problem of deriving hierarchical semantics from social annotations.
2. The proposal of an unsupervised model for automatic semantic clustering, and hierarchical relationship identification.
3. The evaluation of the proposed model on both Del.icio.us and Flickr. The preliminary experimental result demonstrates the model's effectiveness.

In our current work, the evaluation of our model is mainly based on people's intuition and common sense. We would do more detailed evaluation by comparing this hierarchical semantics with other web taxonomy, like ODP. Moreover, we would emphasize on applying our results in real applications to measure our model's efficiency.

Acknowledgement

References

1. Smith, G.: Folksonomy: social classification. Atomiq/Information Architecture [blog] (2004), `http://atomiq.org/archives/2004/08/folksonomy_social_classification.html`
2. Aurnhammer, M., Hanappe, P., Steels, L.: Augmenting navigation for collaborative tagging with emergent semantics. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L. (eds.) ISWC 2006. LNCS, vol. 4273, Springer, Heidelberg (2006)
3. Wu, X., Zhang, L., Yu, Y.: Exploring social annotations for the semantic web. In: Proceedings of the WWW 2006, pp. 417–426 (2006)
4. Li, R., Bao, S., Fei, B., Su, Z., Yu, Y.: Towards effective browsing of large scale social annotations. In: Proceedings of the WWW 2007, pp. 943–952 (2007)

5. Mathes, A.: Folksonomies-cooperative classification and communication through shared metadata. Computer Mediated Communication, LIS590CMC (Doctoral Seminar), Graduate School of Library and Information Science, University of Illinois Urbana-Champaign (December 2004)
6. Quintarelli, E.: Folksonomies: power to the people. ISKO Italy-UniMIB meeting (June 2005), available at `http://www.iskoi.org/doc/folksonomies.htm`
7. Hammond, T., Hannay, T., Lund, B., Scott, J.: Social bookmarking tools (i). D-Lib Magazine 11(4), 1082–9873 (2005)
8. Golder, S., Huberman, B.: Usage patterns of collaborative tagging systems. Journal of Information Science 32(2), 198 (2006)
9. Halpin, H., Robu, V., Shepherd, H.: The complex dynamics of collaborative tagging. In: Proceedings of the WWW 2007, pp. 211–220 (2007)
10. Aberer, K., Cudre-Mauroux, P., Ouksel, A., Catarci, T., Hacid, M., Illarramendi, A., Kashyap, V., Mecella, M., Mena, E., Neuhold, E., et al.: Emergent semantics principles and issues. In: Lee, Y., Li, J., Whang, K.-Y., Lee, D. (eds.) DASFAA 2004. LNCS, vol. 2973, Springer, Heidelberg (2004)
11. Mika, P.: Ontologies are us: a unified model of social networks and semantics. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 522–536. Springer, Heidelberg (2005)
12. Brooks, C., Montanez, N.: Improved annotation of the blogosphere via autotagging and hierarchical clustering. In: Proceedings of the WWW 2006, pp. 625–632 (2006)
13. Dubinko, M., Kumar, R., Magnani, J., Novak, J., Raghavan, P., Tomkins, A.: Visualizing tags over time. In: Proceedings of the WWW 2006, pp. 193–202 (2006)
14. Hotho, A., Jaschke, R., Schmitz, C., Stumme, G.: Information retrieval in folksonomies: Search and ranking. In: Sure, Y., Domingue, J. (eds.) ESWC 2006. LNCS, vol. 4011, Springer, Heidelberg (2006)
15. Bao, S., Wu, X., Fei, B., Xue, G., Su, Z., Yu, Y.: Optimizing web search using social annotations. In: Proceedings of WWW 2007, pp. 501–510 (2007)
16. Pereira, F., Tishby, N., Lee, L.: Distributional clustering of english words. In: Proceedings of the 31st conference on Association for Computational Linguistics, pp. 183–190 (1993)
17. Yang, X.L., Song, Q., Zhang, W.: Kernel-based deterministic annealing algorithm for data clustering. IEEE Proceedings-Vision, Image, and Signal Processing 153, 557 (2006)
18. Wanhyun, C., Park, J., Lee, M., Park, S.: Unsupervised color image segmentation using mean shift and deterministic annealing em. In: ICCSA 2004. Internat. Conf. on Computational Science and Its Applications, vol. 3, pp. 867–876 (2004)
19. Rose, K.: Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. Proceedings of the IEEE 86(11), 2210–2239 (1998)
20. Schachter, J.: Del.icio.us about page (2004), `http://del.icio.us/about/`
21. Vander Wal, T.: Explaining and showing broad and narrow folksonomies (2005), `http://www.personalinfocloud.com/2005/02/explaining_and_.html`