The Ninth International Conference on Web-Age Information Management

Co-Clustering Tags and Social Data Sources

Eirini Giannakidou, Vassiliki Koutsonikola, Athena Vakali
Department of Informatics
Aristotle University
54124 Thessaloniki, Greece
{eirgiann, vkoutson, avakali}@csd.auth.gr

Ioannis Kompatsiaris
Informatics and Telematics Institute
CERTH
Thermi-Thessaloniki, Greece
ikom@iti.gr

Abstract

Under social tagging systems, a typical Web 2.0 application, users label digital data sources by using freely chosen textual descriptions (tags). Poor retrieval in the aforementioned systems remains a major problem mostly due to questionable tag validity and tag ambiguity. Earlier clustering techniques have shown limited improvements, since they were based mostly on tag co-occurrences. In this paper, a co-clustering approach is employed, that exploits joint groups of related tags and social data sources, in which both social and semantic aspects of tags are considered simultaneously. Experimental results demonstrate the efficiency and the beneficial outcome of the proposed approach in correlating relevant tags and resources.

1 Introduction

Social Tagging Systems is a typical, popular and promising Web 2.0 application [13], where users label digital data sources by using freely chosen textual descriptions (tags). The resources along with its accompanying metadata (tags) are available to the entire web community. This user-driven approach of information creation and organization is known as *folksonomy*, a term coined by Th. Vander Wal, to express the resultant categorization scheme, along with its collective nature [17]. As highlighted in [2], [8], folksonomies have structure and dynamics similar to those of a *complex system*, i.e. knowledge is built incrementally in an evolutionary and decentralized manner, yielding stable and knowledge-rich patterns, namely *Emergent Semantics* ([16]). Unlike earlier static knowledge representation structures, folksonomies are dynamic and have a noteworthy ability in capturing the community' s point of view of the specific data sources and the general trends, at a given time. Additionally, they capture social relations between the community members. Therefore, they constitute promising data structures for knowledge mining.

Hence, a nontrivial group of knowledge research community has focused on the exploitation of social data (i.e. folksonomies), achieving limited success, yet. This shortfall of knowledge extraction from social tagging systems originates mostly from the *questionable tag validity*, together with the *flat structure* (lack of hierarchical or other relations) these systems have, which results in tag redundancies and ambiguities [6].

Clustering is often introduced in the bibliography of social tagging systems as an approach to overcome their intrinsic limitations, mentioned above, and derive knowledge regarding their content or their users. The idea is: divide the resources into semantically related clusters (i.e. meaningful groups of resources) and exploit the shared understanding about tags and resources fostered in each cluster. The division is performed according to some *metric of similarity* and each extracted cluster would ideally correspond to a specific topic. The expected benefit of the whole process is that the collective activity of tagging will isolate erroneous tags and illustrate the dominant tags in each cluster, expressing, thus, the community's point of view around the corresponding topic.

More specifically, in [1] the authors demonstrate that clustering enhances user experience in a social tagging system. Additionally, in the Flickr[1] photo-sharing system, the use of Flickr clusters handles quite well the tag ambiguity issue, as the implementation achieves to separate different senses of ambiguous tags in different clusters and fascinates the user exploration inside the system. Other efforts focus on the combination of social and semantic web, so as the ontologies could be dynamic and based on a social ground. These methods utilize clustering, in order to identify tag patterns inside folksonomies and exploit them in ontology extraction or enrichment [14], [12], [15], [18], [20]. All earlier efforts implement clustering based solely on statistical analysis of tag co-occurence. They do not consider at all the semantic aspects of tags, which may cause semantically-

[1] Flickr photo-sharing system: *http://www.flickr.com*

978-0-7695-3185-4/08 $25.00 © 2008 IEEE
DOI 10.1109/WAIM.2008.61

Authorized licensed use limited to: Aristotle University of Thessaloniki. Downloaded on February 17, 2009 at 07:02 from IEEE Xplore. Restrictions apply.

related (e.g. synonyms) tags to be separated in different clusters, because people have not used them together in their annotations. This limitation may lead to the decomposition of a meaningful group into many smaller ones and, thus, the loss of the real relations between tags.

In order for the clustering to be effective and yield pure clusters, an appropriate metric of similarity between the resources needs to be employed. Here, we apply a metric that defines resources' similarity in proportion with their corresponding tags' similarity combining jointly social and semantic aspects of resources accompanying tags, so as to calculate their distance. Tag co-occurrence (i.e. social aspects of tags) is commonly employed as a similarity metric between social data, as described above. Indeed, the fact that a lot of people tend to use some tags together indicates that there is a relation between them. However, applying solely this metric often yields meaningless clusters, which cannot be interpreted and mapped to a particular topic. To this end, semantic knowledge about the tags is also taken into account in their distance estimation.

In this paper, a co-clustering method is utilized that employs the above similarity metric and yields a series of clusters, each of which contains a set of resources together with a set of tags. Co-clustering is proposed as a more suitable approach, which has been used in grouping together elements from different datasets [4], [3]. In our case, co-clustering will be used to relate tags and social data sources.

The proposed method has a number of potential applications. Two are quoted indicatively:

- Ontology enrichment using extracted concepts or relations out of folksonomies. The emergent ontologies will encompass the users' point of view under a certain domain, be open to new trends and embed complex system characteristics

- Training of multimedia processing algorithms (in case of multimedia social data). This process requires extended effort on manually annotating multimedia resources, which can be avoided by exploiting the annotations performed in each folksonomy cluster.

The remainder of the paper is structured as follows. Section 2 describes the basic notation used and defines our problem formulation, while Section 3 analyzes the employed similarity measures and the proposed co-clustering approach. Section 4 provides our experimentation. The conclusions are presented in Section 5.

2 Problem Formulation

A Social Tagging System, STS, is a web-based application, where users assign tags (i.e. arbitrary textual descriptions) to digital resources. The digital resources are either uploaded by users or, are, already, available in the web. The users are either "isolated" or, more commonly, members of web communities (i.e. social networks) and their main motivation (for tagging) is information organization and sharing. The tagging activity inside an STS shows the way users categorize resources and it is known as its folksonomy [17]. Figure 1 depicts the basic structure of a web-based social tagging system.

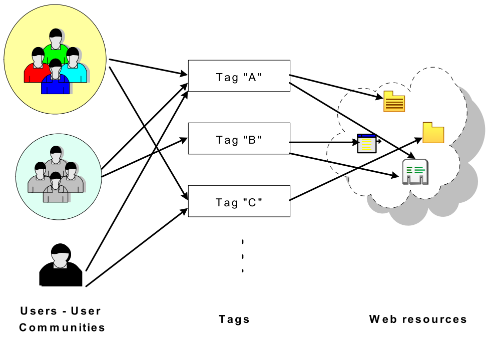


Figure 1. A web-based social tagging system.

We consider an STS and the finite sets U, R, T, A which describe the set of users, resources, tags and user annotations (i.e. tag assignments), respectively. Table 1 summarizes the basic symbols' notation used in this paper.

Table 1. Basic Symbols Notation.

Symbol	Definition
m, n, l, p, d	Number of users, resources, tags, user's annotations and attributes (respectively)
U	Users' set $\{u_1, \ldots, u_m\}$
R	Resources' set $\{r_1, \ldots, r_n\}$
T	Tags' set $\{t_1, \ldots, t_l\}$
A	User's annotation set $\{a_1, \ldots, a_p\}$
AS	Attributes set $\{at_1, \ldots, at_d\}$
$f(r_i, t_j)$	Annotation function of tag t_j to resource r_i

Definition 1 (FOLKSONOMY OF AN STS) *Given a Social Tagging System (STS), its derived folksonomy* $\boldsymbol{F}$ *is defined as the tuple* $\boldsymbol{F} = (U, R, T, A)$*, where* $A \subseteq U \times R \times T$*, i.e. the users' annotation set* A *is modeled as a triadic relation between the other sets.*

The above definition was initially introduced in [9] and is also adopted in our approach.

Each STS handles a particular type of resources. For instance, Flickr handles photos, while del.icio.us [2] handles urls, YouTube [3] handles videos, etc. Nevertheless, resource

[2] del.icio.us social bookmarking system: *http://del.icio.us*

[3] YouTube video broadcast: *http://www.youtube.com*

Authorized licensed use limited to: Aristotle University of Thessaloniki. Downloaded on February 17, 2009 at 07:02 from IEEE Xplore. Restrictions apply.

management by an STS is a transparent process, which does not rely on the varying nature of digital resources, but involves only their user-generated metadata (produced through the tagging activity). In this paper we consider that the context of each resource is captured by the annotations (i.e. group of tags) it has received. Therefore, we define an annotation function f to determine whether a tag $t_j, j = 1 \ldots l$, has been used for the annotation of resource $r_i, i = 1 \ldots n$, as follows:

$$f(r_i, t_j) = \begin{cases} 1 & \textit{if } t_j \textit{ is an annotation tag for } r_i \\ 0 & \textit{otherwise} \end{cases}$$

We can now characterize and define resources considering their corresponding tags.

Definition 2 (RESOURCE'S REPRESENTATION) *Each resource $r_i \in R$, where $i = 1 \ldots n$, is represented by aggregating the tags assigned to it by all users and it is identified by:*

$$r_i = \{\cup t_x\}, \forall t_x \in T : f(r_i, t_x) = 1$$

In practice, the number of tags used to represent a specific resource may grow in large scale and thus we need to employ a selection process of the most distinguishing tags which will form the resources' attribute set AS. In our approach we use the d most frequent tags to form the AS set, which will guide our clustering process.

Definition 3 (THE ATTRIBUTE SET) *Given the $T = \{t_1, \ldots, t_l\}$ set of tags, we define the attribute set $AS = \{at_1, \ldots, at_d\}$: $AS \subseteq T$ and AS contains the d most frequent tags $t_x \in T$.*

Each attribute $at_y \in AS$ is related with a different degree to the various r_i, $1 \leq i \leq n$, resources while two different resources may be indirectly related, if they present strong relation with the same set of attributes. The relation between two resources is based on both social and semantic aspects of their involving tags.

Our purpose is to create groups of related resources and attributes and, thus, we need to provide solution to the following RESOURCES-ATTRIBUTES CO-CLUSTERING problem.

Problem 1 (RESOURCES-ATTRIBUTES CO-CLUSTERING) *Given a set R of n resources, a set AS of d attributes, an integer k and a $Similarity$ function, find a set C of k subsets of both resources and attributes, $C = \{C_1, \ldots, C_k\}$ such that $\sum_{x=1}^{k} \sum_{r_i, at_j \in C_x} Similarity(r_i, at_j)$, $i = 1, \ldots, n$ and $j = 1, \ldots, d$, is maximized* .■

The $Similarity$ function must be defined in a way to sufficiently capture the association between each resource and each attribute by jointly considering the social and semantic aspects of the involved tags and attributes.

3 Clustering STS Resources and Attributes

3.1 Capturing similarities

As already discussed in Section 2, each resource is represented by the set of tags that have been used for its annotation (Definition 2). Thus, finding the relation between a resource and an attribute indicates capturing the similarity between the resources' tags and the attribute. Existing approaches are based solely on the tagging co-occurrence information which is captured by the so-called *Social Similarity*. We define the *Social Similarity* between two tags t_x and t_y, where $1 \leq x, y \leq l$ as follows:

$$SoS(t_x, t_y) = \frac{\sum_{i=1}^{n} r_i : (u_w, r_i, t_x) \in A \textit{ and } (u_z, r_i, t_y) \in A}{\max(\sum_{i=1}^{n} r_i : (u_w, r_i, t_x) \in A, \sum_{i=1}^{n} r_i : (u_z, r_i, t_y) \in A)} \quad (1)$$

where $u_w, u_j \in U$, $r_i \in R$.

However, considering the semantic aspect of tags, as well, is expected to be beneficial for the clustering process in an STS, since it can contribute to eliminating the tag synonymy issue. For the estimation of the *Semantic Similarity* between two tags, we need to use external resources (i.e. web ontologies, thesauri, etc) and a mapping technique between tags and the resource's concepts. In our work, we adopted the approach described in [19], due to its straightforward application to our data, according to which the semantic distance between two concepts is proportional to the path distance between them. For example, let t_x and t_y be two tags for which we want to find the semantic similarity and $\overrightarrow{t_x}$, $\overrightarrow{t_y}$ be their corresponding mapping concepts via an ontology. Then, their *Semantic Similarity* SeS is calculated as:

$$SeS(t_x, t_y) = \frac{2 \times depth(LCS)}{[depth(\overrightarrow{t_x}) + depth(\overrightarrow{t_y})]} \quad (2)$$

where $depth(\overrightarrow{t_x})$ is the maximum path length from the root to $\overrightarrow{t_x}$ and *LCS* is the least common subsumer of $\overrightarrow{t_x}$ and $\overrightarrow{t_y}$.

The total similarity between two tags will be estimated by considering both their social and semantic similarity (Equations 1, 2). In order to examine the impact that each kind of information has on the clustering process, we combine them in the form of a weighted sum. Specifically, a factor w is employed to define the effect each track has on the estimation of their joint similarity. Thus, we define the *Similarity Score SS* between two tags t_x and t_y in terms of both their social (Equation 1) and semantic (Equation 2) similarity as:

$$SS(t_x, t_y) = w * \textit{SoS}(t_x, t_y) + (1 - w) * \textit{SeS}(t_x, t_y) \quad (3)$$

where $w \in [0, 1]$. To this context, when $w = 1$ we consider solely the *Social Similaty SoS*, while when $w = 0$, only the *Semantic Similarity SeS* is considered. For any other value

Authorized licensed use limited to: Aristotle University of Thessaloniki. Downloaded on February 17, 2009 at 07:02 from IEEE Xplore. Restrictions apply.

of w both similarities contribute to the *Similarity Score SS* of two tags.

Given the *Similarity Score* (Equation 3) between two tags, we proceed to the definition of the *Similarity* function between a resource r_i, which is represented as a set of tags assigned by users to the resource (Section 2), and an attribute at_j. The *Similarity* function is the maximum Similarity Score between every tag assigned to the resource r_i and the attribute at_j. Thus:

$$Similarity(r_i, at_j) = max_{x=1 \ldots |r_i|}\{SS(t_x, at_j)\} \quad (4)$$

where $r_i \in R$, $t_x \in r_i$, $at_j \in AS$.

The values of Similarity function between each of the n resources and d attributes are then used to form the $n \times d$ table RA as follows:

$$RA(i, j) = Similarity(r_i, at_j) \quad (5)$$

where $i = 1, \ldots, n$ and $j = 1, \ldots, d$.

3.2 Dataset Representation

Applying a typical clustering algorithm to *RA* table (Equation 5) would yield clusters with elements from only one dataset. Since our problem deals with the simultaneous clustering of both resources and attributes, we need to use a data structure that will efficiently represent the two datasets elements along with their relations and at the same time it will enable the co-clustering process. A graph is such a convenient structure, since it can represent the relations between the resources and attributes and has already been used in co-clustering approaches [4]. In our case we consider a bipartite graph with its vertices indicating the resources and attributes and its edges representing the calculated relations using the $Similarity$ function (Equation 4). Let us consider the bipartite graph $G = (R, AS; E)$ presented in Figure 2 where $R = \{r_1, r_2, r_3\}$ the set of resources, $AS = \{at_1, at_2, at_3, at_4\}$ the set of attributes and $E = \{\{r_i, at_j\} : r_i \in R, at_j \in AS\}$ the set of edges connecting resources and attributes. Each of the edges expresses the relation between the connected resource r_i and attribute at_j and its weight is equal to $Similarity(r_i, at_j)$. According to our problem definition we aim to create k subsets $C_1, C_2, \ldots, C_k$ of elements containing both resources and tags and resulting in the maximization of $\sum_{x=1}^{k} \sum_{r_i, at_j \in C_x} Similarity(r_i, at_j)$, $i = 1, \ldots, n$ and $j = 1, \ldots, d$. In case of the bipartite graph of Figure 2, its 2-partitioning depicted in Figure 3 would result in the maximization of the sum of similarities between the elements belonging to the same cluster, while the sum of similarities between the elements of different clusters would be minimized. In other words, we are looking for a k-partitioning of the graph, such that

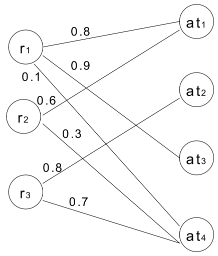


Figure 2. Data representation.

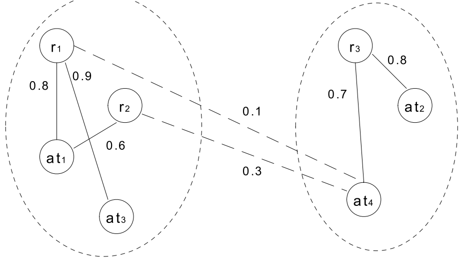


Figure 3. Cut of the Bipartite Graph.

$$\sum_{r_i \in C_x} \sum_{at_j \in C_y} Similarity(r_i, at_j), \quad (6)$$

where $x, y = 1 \ldots k$ and $x \neq y$, is minimized.

The last quantity corresponds to the *cut* of the graph G and thus our Problem 1 is transformed into a graph k-partitioning.

3.3 The Co-Clustering Algorithm

Using a graph to represent our datasets' elements and their relations motivated us to follow the principles of spectral graph theory, which have been successfully applied in graph partitioning problems [7], [11]. Spectral graph clustering algorithms rely on the eigenstructure of a similarity matrix to partition points into disjoint clusters, with points in the same cluster having high similarity and points in different clusters having low similarity. More specifically, an eigenvector decomposition is performed on the similarity matrix and then, traditional clustering techniques, such as K-means, may be a applied to the subspace defined by the eigenvectors.

The similarity matrix which describes our weighted graph G is the *RA* table (Equation 5). As it has been proven in [4] the k left and right singular vectors of the normalized table $NRA = D_r^{-1/2} RA D_{at}^{-1/2}$ provide a real approximation to the discrete graph k-partitioning problem. The D_r

Authorized licensed use limited to: Aristotle University of Thessaloniki. Downloaded on February 17, 2009 at 07:02 from IEEE Xplore. Restrictions apply.

and D_{at} are the diagonal degree tables of resources and attributes, respectively, and they are defined as follows:

$$D_r(i,i) = \sum_{j=1}^{d} RA(r_i, at_j), i = 1, \ldots, n$$

$$D_{at}(j,j) = \sum_{i=1}^{n} RA(r_i, at_j), j = 1, \ldots, d$$

Let L_r denote the n x k table of the left singular vectors and R_{at} the d x k table of the right singular vectors of *NRA* table. In order to perform a simultaneous clustering of r_i, $i = 1, \ldots, n$, and at_j, $j = 1, \ldots, d$, elements, we create the (n + d) x k two dimensional table *SV* defined as:

$$SV = \left[\begin{array}{c} D_r^{-1/2} L_r \\ D_{at}^{-1/2} R_{at} \end{array} \right]$$

Running a typical clustering algorithm on SV will result in k clusters containing elements from both resources and attributes sets. In the first step of the CO-CLUSTERING algorithm, a data preprocessing (line 2) takes place where a filtering of the tags is applied in order to result in more meaningful, for the clustering process, tags. More specifically, the preprocessing involves two steps. In the first one, a spelling normalization occurs, so that different written forms of the same tag are mapped to the same normalized tag (e.g. Sea, sea). Then, the infrequent tags which lack a proper meaning and cannot be mapped to any real concepts are filtered out as noise. Indeed, since both the semantic and social knowledge regarding these tags are negligible, they can be left out of the clustering process, with no considerable lack of information. The preprocessing step results in the T^* set of attributes where $T^* \subseteq T$. Given the T^* set, the algorithm computes and extracts the d most frequent tags in order to form the attributes set *AS* (line 3), as described in Section 2. Then, the social (Equation 1) and semantic (Equation 2) similarities between tags are calculated (lines 5 and 6) and are used to find tag similarity scores *SS* (Equation 3) (line 7). The factor w weights the significance of the social and semantic similarities. The similarities between resources and attributes are estimated by the Similarity function (Equation 4) and are stored in the two dimensional $n \times d$ table RA (line 8). Once the *RA* is created, we proceed to the co-clustering step. We calculate the degree tables D_r and D_{at} (line 10) and then we form the *NRA* table (line 11) on which we apply a singular value decomposition to obtain the k left and right singular vectors which are organized in tables L_r and D_{at} (line 12), respectively. D_r, D_{at}, L_r and R_{at} are integrated in the *SV* table (line 13) on which we run k-means clustering algorithm. The algorithm finalizes with the k obtained clusters which contain both resources and attributes (line 14).

Algorithm 1 The CO-CLUSTERING algorithm.

```
Input: The set $R$ of $n$ resources, the set $T$ of $l$ tags and two
    integers $k$ and $w$ where $w \in [0..1]$
Output: A set $C = \{C_1, \ldots, C_k\}$ of $k$ subsets consisting
    of elements from both $R$ and $T$, such that the sum of
    inter-clusters similarities defined by (6) is minimized.
1: /*Preprocessing*/
2: $T^* = Preprocess(T)$
3: $AS = ExtractAttributes(T^*)$
4: /*capturing similarities*/
5: $SoS = CalculateSocialSimilarity(R, AS)$
6: $SeS = CalculateSemanticSimilarity(R, AS)$
7: $SS = w * SoS + (1 - w) * SeS$
8: $RA = Similarity(SS)$
9: /*Co-clustering process*/
10: $(D_r, D_{at}) = ComputeDegreeTables(RA)$
11: $NRA = D_r^{-1/2} RA D_{at}^{-1/2}$
12: $(L_r, R_{at}) = SVD(NRA)$
13: $SV = CreateIntegratedTable(D_r, D_{at}, L_r, R_{at})$
14: $C = k - means(SV, k)$
```

4 Experimentation

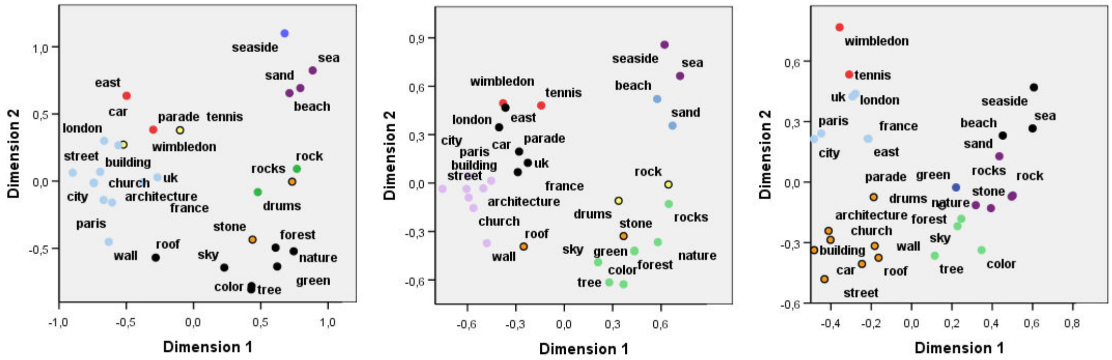
To carry out the experimentation phase and the evaluation of the proposed clustering approach, a dataset from Flickr was crawled using wget[4] utility and Flickr API facilities. It consists of 3000 images depicting cityscape, seaside, mountain, roadside, landscape, sport-side and locations (about 500 images from each domain). As a source of semantic information for tag concepts, we employ the lexicon WordNet [5], which stores english words organized in hierarchies, depending on their cognitive meaning. After the preprocessing phase, the attribute set AS was extracted. In the experimentation that follows, we restricted the size of AS to d = 30 tags, in order to facilitate the graphical demonstration.

4.1 Attribute Assignment Interpretation

In this first section of our experimentation, the cluster assignment of attributes is examined. Specifically, we study the impact of the weight factor w on the clustering results. It is reminded that w defines the affect of *Social* and *Semantic Similarity* on the extracted clusters. We consider the following three cases: i)w = 0.2, in which *Social Similarity* (i.e. tag-co-occurence) is favored, ii) w = 0.5, in which both kinds of similarity are equally taken into account, and, iii) w = 0.8, in which the *Semantic Similarity* (i.e. actual meaning of tags) is given advantage. In order to proceed to a "conceptual" analysis of the clustering results, we per-

[4] wget: *http://www.gnu.org/software/wget*

Authorized licensed use limited to: Aristotle University of Thessaloniki. Downloaded on February 17, 2009 at 07:02 from IEEE Xplore. Restrictions apply.



(a) $w = 0.2$

(b) $w = 0.5$

(c) $w = 0.8$

Figure 4. Attributes distribution to $k = 8$ clusters.

formed a correspondence analysis, which allows the visualization of the extracted associations between attributes and their final assignment in the obtained clusters. Figure 4 depicts the results of the correspondence analysis, in each of the aforementioned three cases of w, using number of clusters $k = 8$ (attributes having the same color belong to the same cluster). As depicted in the aforementioned figure, the CO-CLUSTERING algorithm manages to identify groups of attributes that appear to be near in terms of their similarity. For example, in all three subfigures *tennis* belongs to the same cluster with *wimbledon* as well as *forest* is grouped with *tree*, while *church* is grouped with *architecture*, independently of the w value. However, changing the value of w results in clusters of different membership, since w may favor *Social* or *Semantic* similarity between attributes.

In case of $w = 0.2$, where more weight is given to the *Social Similarity*, we can derive that the attributes assigned by the algorithm in the same cluster are attributes that co-occur in the users' annotations (Figure 4(a)). For example the attributes *forest*, *nature*, *green*, *tree* belong to the same cluster, because these tags are often used together for describing images related to sceneries of nature. The same holds for the cluster where *street*, *building*, *church*, *architecture* are assigned, since they constitute tags that occur frequently in the description of images referring to city places. It is worth noticing that the attributes *rock* and *rocks* are assigned to different clusters (*rock* is grouped with *drums*, while *rocks* is grouped with *stone*), which indicates that in most annotations the tag *rock* is used in terms of the well-known type of music rather than in the sense of stone. In general, tag co-occurence has proven to be more advantageous in the case of ambiguous tags (homonyms), since it is the context of such a tag (i.e. its co-occuring tags) that will help to disambiguate its meaning. However, lacking semantic information, the algorithm splits meaningful clusters into subclusters (i.e. *sea*, *beach*, *sand* are assigned in one cluster, while *seaside* is assigned to another).

When $w = 0.5$, both *Social* and *Semantic* similarities are equally considered and now the algorithm is more possible to group attributes that semantically are close. As can be seen in Figure 4(b), the *sea* is now grouped with *seaside*, since these two attributes are semantically close. Although the *Semantic similarity* is taken more into account (compared to $w = 0.2$), the fact that *Social Similarity* is fairly considered prevents the algorithm from assigning *rocks* and *rock* to the same cluster.

For $w = 0.8$, Figure 4(c), where the *Semantic Similarity* is favored, the algorithm assigns all related attributes from the *sea* domain in one cluster i.e. *sea*, *seaside*, *beach*, *sand* (*beach* and *sand* overlap). Despite the fact that all aforementioned tags are closely akin, in the previous described cases, they are split into different clusters, due to the fact that the users have not used all of them together in their annotations. However this method fails in disambiguating correctly the *rock* tag, as it assigns *rock* and *rocks* to the same cluster even though in most cases they are not used in the same sense and they do not describe the same set of images. Moreover, as we can see, *drums* is not clustered with any other attribute because no other attribute is semantically related to it. Thus, we can conclude that while this approach yields semantically meaningful clusters around a specific topic and it tackles well in case of synonyms (or tags with alike meaning), it fails to handle the tag ambiguity issue.

From the above discussion it is clear that the value we choose for w affects the way the CO-CLUSTERING algorithm groups attributes and hence resources (images). A similar analysis for resources is not feasible due to their high number (3000), which would result in not so readable visualizations. Therefore, we need to proceed to a qualitative evaluation of the clusters.

Authorized licensed use limited to: Aristotle University of Thessaloniki. Downloaded on February 17, 2009 at 07:02 from IEEE Xplore. Restrictions apply.

4.2 Clustering Results Evaluation

In this section, the overall evaluation of the obtained clusters is examined. First, precision and recall measures are used, to show us whether resources belonging to the same cluster are relative to each other in terms of their tags. Let R_{C_j} and AS_{C_j} denote the set of resources and the set of attributes that have been assigned to cluster C_j, respectively. Then, the precision and recall of the cluster C_j are defined as follows:

$$Precision(C_j) = \frac{\sum_{i=1}^{|R_{C_j}|} r_i : r_i \cap AS_{C_j} \neq \emptyset, \forall r_i \in C_j}{|C_j|}$$

$$Recall(C_j) = \frac{\sum_{i=1}^{|R_{C_j}|} r_i : r_i \cap AS_{C_j} \neq \emptyset, \forall r_i \in C_j}{\sum_{i=1}^{|R|} r_x : r_x \cap AS_{C_j} \neq \emptyset, \forall r_x \in R}$$

We experimented with the indicative values of $k = 8$ and $k = 10$, while we set w to $= 0.2, 0.5$ and 0.8, (as described in the previous section). The calculated precision and recall values for each of the obtained clusters are depicted in Table 2 and 3, respectively. The results presented in Table 2 are generally better than the ones in Table 3, showing that in most cases the extracted clusters are pure (high precision) but sometimes meaningful clusters are split (low recall). Moreover, as can been seen in both tables, the value of w affects the results. More specifically, we observe that, in most cases, for $w = 0.5$ both precision and recall have their highest values, meaning that the incorporation of both kinds of knowledge is more advantageous towards relying solely on one of them. Thus, our method is expected to outperform existing approaches (quoted in Section I), since most of them are based on social similarity, which is equivalent to minimizing the affect of w in our case.

Table 2. Clusters' Precision.

w	Cluster ($k = 8$)									
	1	2	3	4	5	6	7	8		
0.2	0.94	0.70	0.81	0.74	0.29	0.96	0.58	0.65		
0.5	0.72	1	0.94	0.51	0.95	0.78	0.61	0.75		
0.8	0.86	0.65	0.31	0.91	0.86	0.49	0.63	0.46		
w	Cluster ($k = 10$)									
	1	2	3	4	5	6	7	8	9	10
0.8	0.52	0.51	1	0.77	0.43	1	0.46	0.52	0.50	0.71
0.5	0.95	0.46	0.70	0.44	0.61	0.47	0.77	0.89	0.93	0.43
0.2	0.57	0.85	0.48	0.76	0.70	0.72	0.74	0.76	0.77	0.79

The ideas of precision and recall are combined in F-Measure which is a broadly accepted and reliable index used in various clustering evaluation approaches [10]. Given the Precision and Recall definitions described in this section, the value of F-measure for a cluster C_j is defined as:

$$F(C_j) = \frac{2 * Precision(C_j) * Recall(C_j)}{Precision(C_j) + Recall(C_j)}$$

Table 3. Clusters' Recall.

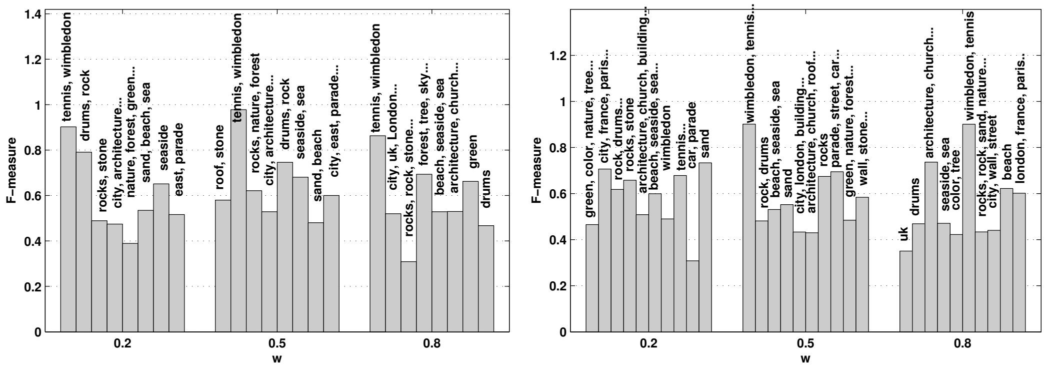
w	Cluster ($k = 8$)									
	1	2	3	4	5	6	7	8		
0.2	0.87	0.91	0.35	0.35	0.58	0.37	0.75	0.43		
0.8	0.49	0.96	0.46	0.54	0.61	0.61	0.39	0.50		
0.8	0.87	0.43	0.31	0.56	0.38	0.57	0.70	0.47		
w	Cluster ($k = 10$)									
	1	2	3	4	5	6	7	8	9	10
0.2	0.39	0.61	0.88	0.58	0.40	0.51	0.37	0.62	0.19	0.69
0.5	0.86	0.50	0.43	0.75	0.34	0.40	0.60	0.57	0.33	0.92
0.8	0.26	0.44	0.58	0.34	0.42	0.82	0.41	0.38	0.81	0.52

The values of F-measure fluctuate in the interval $[0..1]$ with higher values indicating a better clustering. Figures 5(a) and 5(b) present F-measure value for each of the obtained clusters for $k = 8$ and $k = 10$, respectively. The value of w was set to 0.2, 0.5 and 0.8. Moreover, indicative attributes of each cluster, that correspond to the extracted cluster's topic, are depicted above each bar. As can be seen, F-measure performance varies between clusters, depending on the topic of each one. We claim that this is analogous to the nature and the number of the attributes assigned to each cluster, as well as their ability to represent thoroughly the extracted topic. For example, in the tennis cluster, the contained attributes (i.e. *tennis*, *wimbledon*) were representative in the particular dataset, since they aggregated most of the relevant resources in the same cluster, in all tested cases. On the contrary, in the case of $w = 0.8$, we see that the grouping of the ambiguous attribute *rock* with *stones* causes irrelevant data sources (i.e. images depicting music themes with ones that show rocky landscapes) to be in the same cluster, thus, deteriorating the algorithm performance. We claim that a better attribute selection method would improve the overall algorithm performance and we plan to do this, as future work. Regarding the w values, the $w = 0.5$ still yields better clusters, comparatively with the other two values.

5 Conclusions

This paper introduces a co-clustering approach for social data grouping that aims to improve the efficiency of tagging systems. The CO-CLUSTERING algorithm considers the semantic in addition to the social aspect of resources accompanying tags in a balanced way and yields clusters consisting of both resources and user annotation tags. The proposed approach has been evaluated under real workload and the results proved its efficiency in correlating relevant tags and resources, illustrating the dominant tags in each cluster and expressing users' point of view around the corresponding topic. Moreover, the consideration of the semantic aspect of user annotation tags enables the CO-CLUSTERING algorithm to handle the tag ambiguity issue. The proposed approach has a number of potential applications in retrieval

Authorized licensed use limited to: Aristotle University of Thessaloniki. Downloaded on February 17, 2009 at 07:02 from IEEE Xplore. Restrictions apply.



(a) $k = 8$

(b) $k = 10$

Figure 5. Clusters' F-measure.

systems, semantics extraction and knowledge mining in general and more specifically in automated multimedia content analysis, being used, for example, as training sets for specific concepts represented by tags. Future work involves improvement of the attributes selection process and experimentation with more attributes and resources.

References

[1] G. Begelman, P. Keller, and F. Smadja. Automated tag clustering: Improving search and exploration in the tag space. *Proc. of the Collaborative Web Tagging Workshop, 15th Int. World Wide Web Conf., (WWW'06)*, pages 89–98, May 2006.

[2] C. Cattuto, V. Loreto, and L. Petronero. Semiotic dynamics and collaborative tagging. *Proc. of the National Academy of Sciences*, 104:1461–1464, January 2007.

[3] I. Dhillon, S. Mallela, and D. Modha. Information-theoretic co-clustering. *Proc. of the 9th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, (KDD'03)*, pages 89–98, August 2003.

[4] I. S. Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. *Proc. of the 7th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data mining, (KDD'01)*, pages 269–274, August 2001.

[5] C. Fellbaum. *WordNet, an electronic lexical database*. The MIT Press, 1990.

[6] S. Golder and A. Huberman. The structure of collaborative tagging systems. May 2006. http://www.hpl.hp.com/research/idl/papers/tags/.

[7] L. Hagen and A. Kahng. New spectral methods for ratio cut partitioning and clustering. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 11(9):1074–1085, 1992.

[8] H.Halpin and H. Shepard. Evolving ontologies from folksonomies: Tagging as a complex system. Complex Systems Summer School Project, Apr. 1990. http://www.ibiblio.org/ hhalpin/homepage/ notes/ taggingcss.html.

[9] A. Hotho, R. Jaschke, C. Schmitz, and G. Stumme. Information retrieval in folksonomies: Search and ranking. *Proc. of 2006 European Semantic Web Conf. , (ESWC'06)*, pages 411–426, June 2006.

[10] B. Larsen and C. Aone. Fast and effective: Text mining using linear-time document clustering. *Proc. of 5th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, (KDD'99)*, pages 16–22, August 1999.

[11] S. Loong and S. Mishra. Spectral graph partitioning analysis in vitro synthesized rna structural folding. *Proc. of 2006 Int. Workshop on Pattern Recognition in Bioinformatics , (PRIB'06)*, pages 81–92, August 2006.

[12] P. Mika. Ontologies are us: A unified model of social networks and semantics. *Proc. of the 4th Int. Semantic Web Conf., (ISWC'05)*, pages 522–536, Novemebr 2005.

[13] T. O'Reilly. What is web 2.0, September 2005. http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/ 30/ what-is-web-20.html.

[14] P. Schmitz. Inducing ontology from flickr tags. *Proc. of the Collaborative Web Tagging Workshop, 15th Int. World Wide Web Conf.*, May 2006.

[15] L. Specia and E. Motta. Integrating folksonomies with the semantic web. *Proc. of the 4th European Semantic Web Conf., (ESWC '07)*, pages 624–639, June 2007.

[16] L. Steels. Semiotic dynamics for embodied agents. *IEEE Intelligent Systems*, 21:32–38, May/June 2006.

[17] T. Vander-Wal. Explaining and showing broad and narrow folksonomies, February 2005. http://www.vanderwal.net/random/category.php?cat=153.

[18] X. Wu, L. Zhang, and Y. Yu. Exploring social annotations for the semantic web. *Proc. of the 15th Int. Conf. on World Wide Web, (WWW '06)*, pages 417–426, May 2006.

[19] Z. Wu and M. Palmer. Verm semantics and lexical selection. *Proc. of the 32nd annual meeting of the association for computational linguistics*, pages 133–138, June 1994.

[20] M. Zhou, S. Bao, X. Wu, and Y. Yu. An unsupervised model for exploring hierarchical semantics from social annotations. *Proc. of the 6th Int. Semantic Web Conf., (ISWC '07)*, pages 680–693, November 2007.

Authorized licensed use limited to: Aristotle University of Thessaloniki. Downloaded on February 17, 2009 at 07:02 from IEEE Xplore. Restrictions apply.