Integrating Folksonomies with the Semantic Web

Lucia Specia and Enrico Motta

Knowledge Media Institute – The Open University
Walton Hall, MK7 6AA, Milton Keynes, UK
{L.Specia, E.Motta}@open.ac.uk

Abstract. While tags in collaborative tagging systems serve primarily an indexing purpose, facilitating search and navigation of resources, the use of the same tags by more than one individual can yield a collective classification schema. We present an approach for making explicit the semantics behind the tag space in social tagging systems, so that this collaborative organization can emerge in the form of groups of concepts and partial ontologies. This is achieved by using a combination of shallow pre-processing strategies and statistical techniques together with knowledge provided by ontologies available on the semantic web. Preliminary results on the del.icio.us and Flickr tag sets show that the approach is very promising: it generates clusters with highly related tags corresponding to concepts in ontologies and meaningful relationships among subsets of these tags can be identified.

1 Introduction

Describing resources by means of a set of keywords is a very common way of organizing content for future use, including search and navigation. A collaborative form of this process for shared web-based resources, called "tagging", or "social tagging / annotation", has been gaining impressive popularity among web users. In the light of the Web 2.0 philosophy, the several social tagging systems available nowadays enable users to annotate their resources (web pages, images, videos, etc.) with a set of words, the so-called "tags", which they believe to be relevant to characterize the resource according to their own needs, without relying on a controlled vocabulary or a previously defined structure. The main goal of this annotation is to facilitate the access to resources and, since the systems allow users to share their resources and annotations, tags serve as links to resources annotated both by their owners and by other users. This allows the emergence of a shared and evolving classification structure, which is sometimes called "folksonomy"[1], i.e., a folk taxonomy, or a lightweight conceptual structure created by the users.

Social tagging systems such as Flickr (http://www.flickr.com/), for photo-sharing, and del.icio.us (http://del.icio.us/), for social bookmarking, are becoming more and more popular, covering nowadays a wide range of resources and communities, with a huge number participants sharing and tagging a large number of resources. For example, del.icio.us is said to have more than 1,000,000 registered users in September 2006, who have been posting more than 100,000 bookmarks each day (http://deli.ckoma.net/stats).

Tagging systems are constituted by three main elements: users, resources and tags. Although in most of the systems tags are not mandatory, they are certainly a very important

[1] As defined by T. Vander Wal (http://www.vanderwal.net/random/entrysel.php?blog=1750)

element. Besides establishing a relationship between a resource and a concept in the user's mind, tags can be thought of as the connecting element between resources and users, with these connections defining (even implicitly) relationships amongst users (several users may use the same tags) and amongst resources (resources can be tagged with the same words).

Taking subsets of the tags in Flickr and del.icio.us as examples, in this paper we focus on the relationships amongst tags themselves and their mapping into formal concepts in ontologies. We are therefore primarily interested in the collective purpose of the tags assigned to resources. In that sense, one of the greatest strengths of the social tagging systems, the fact that no pre-defined vocabulary is assumed, leads to a number of limitations and weaknesses in what concerns the use of the tags to retrieve content. As highlighted by Golder and Huberman (2005), the main problems of social tagging systems include *ambiguity*, *lack of synonymy* and *discrepancies in granularity*. An ambiguous word, e.g. `apple`, may refer to the `fruit` or the `computer company`, and this in practice can make the user retrieve undesired results for a certain query. Synonyms like `lorry` and `truck`, or the lack of consistency among users in choosing tags for similar resources, e.g., `nyc` and `new york city`, makes it impossible for the user to retrieve all the desired resources unless he/she knows all the possible variants of the tags that may have been used. Different levels of granularity in the tags may also be a problem: documents tagged `java` may be too specific for some users, but documents tagged `programming` may be too general for others.

We present an approach to minimize these problems by making explicit the semantics behind the tag space in social annotation systems. This is achieved by a pipeline of processes including the cleaning up of tags, the analysis of co-occurrence among tags, the clustering of tags based on the co-occurrence information, and finally the mapping of tags in a cluster into elements (concepts, properties or instances) in ontologies and the extraction of (taxonomic or non-taxonomic) semantic relations between them, using for that information from ontologies available on the semantic web, as well as resources like Wikipedia and Google. Although other attempts have been made to bring the semantics of folksonomies to the surface, as we discuss in Session 2, they do not go beyond finding groups of related tags - no assumption is made about the nature of the tags or the relationships within clusters. Moreover, a systematic evaluation on the quality of the clusters is not performed.

As result of our approach, we obtain groups of highly related tags corresponding to elements in ontologies, structured according to the relationships holding amongst those elements, which can be thought of as *faceted ontologies*, that is, partial ontologies conceptualizing specific facets of knowledge. In contrast with traditional "monolithic" ontologies, the resulting ontologies are constructed by putting together fragments derived from multiple ontologies on the semantic web[2]. These resulting ontologies can be used to enhance various tasks in the tagging systems (for users and semantic web applications):

a) **Query (tag) extension/disambiguation**: in searches for tags, query extension to all related tags (or a subset of them) can be offered to the user. Simple heuristics, based on the types of tags, can be used to extend the search to a subset of the related tags. Also, if the searched tag is ambiguous, the user can be given the related tags in each cluster it appears in order to choose the sense. The search can then be restricted to that sense by adding another tag (from the cluster) to the query.

[2] In (Motta and Sabou, 2006) we argue that this ability of dynamically combining and integrating information coming from multiple ontologies on the web is one of the key features of the emerging new generation of semantic web applications.

b) **Visualization**: clusters of related tags (and the relationships among them) can be graphically presented to provide a better understanding on the way the searched tag is used in the system, which can also be used for query extension / disambiguation.
c) **Tag suggestion**: when tagging a resource, the user can be offered suggestions of "good" tags, based on other tags used by other people for that resource (like in del.icio.us), or related tags that are highly frequent in a given cluster.

The approach can also be used to support **ontology evolution and population**: the new and dynamic knowledge provided by users can complement the formal knowledge in ontologies by adding concepts (or instances of concepts) and relationships (or instances of relationships) between concepts in that ontology. Therefore, with our approach to integrating folksonomies and the semantic web we intend to show ultimately both (i) that the ontologies provided by the semantic web can be used to structure folksonomies semantically and (ii) that the dynamic knowledge provided by folksonomies can be used as a resource for bottom-up knowledge acquisition to support ontology evolution.

The rest of this paper is organized as follows. In Section 2 we describe a few approaches that are related to our work. In Section 3 we present our approach to integrate folksonomies with the semantic web. In Section 4 we show the results of initial experiments with Flickr and del.icio.us tag sets. We conclude with some remarks and future work in Section 5.

2 Related Work

Because one important step in our work is the identification of relations between tags, before comparing our work to other approaches that try to extract semantics from tagging systems, it is important to distinguish it from traditional approaches to relation extraction from texts (Schutz and Buitelaar, 2005; Specia and Motta, 2006). The main difference is that here we cannot count on the conventional notion of "context", i.e., surrounding words around the tag. Some attempts have been made to use information about the resources as context, but there is no guarantee that this *ad hoc* context will offer helpful clues. For example, (Aurnhammer et al., 2006) uses image content features as context to improve search in Flickr: an ordinary search by tag is accomplished and the user then selects a subset of the resulting images to perform another search by "similar" images according to two simple features - colour and texture. Images with other tags, not necessarily similar to the initial one, can therefore be retrieved. However, this image retrieval strategy is unlikely to work well with complex images. In our approach, we rely on no additional context except the tags themselves.

Aiming to induce faceted ontologies from Flickr tags, (Schmitz, 2006) uses a subsumption-based model, derived from the co-occurrence of tags, to find candidate subsumption relations: a tag *x* subsumes another tag *y* if the probability of *x* occurring given *y* is above a certain threshold and the probability of *y* occurring given *x* is below that same threshold. Given the resulting set of "candidate pairs of tags", a tree of possible parent-child relationships is built, with certain candidate pairs being filtered out according to their position and thus reinforcing the remaining relationships. For each leaf of the tree, the best path to the root is chosen accordingly and partial paths are merged into sub-trees. Some of the illustrated sub-trees show that common features hold amongst certain tags. For example, a resulting tree contains `san francisco` as the subsuming tag and a set of children like `civiccenter`, `cliffhouse`, `streetfair`, `muni`. From a semantic point of view this approach is however limited, as in the general case the identified relationships will vary

considerably (e.g., these trees may mix type-of, hyponym, or part-of relationships), but these distinctions are not captured.

Other approaches that concentrate in finding groups of potentially related tags include those of (Begelman et al., 2006) and (Wu et al., 2006). In (Begelman et al., 2006), the tag space is first organized according to their co-occurrences in annotating different resources. A cutoff co-occurrence value is defined based on disruption points in frequency graphs. This new tag space is then represented as an undirected graph, having strongly related tags as vertices and edges with pairs of tags weighted according to the number of times they co-occur. This yields clusters of related tags, but since some clusters are very big, a spectral clustering algorithm is applied to refine them. Amongst the illustrated examples of clusters created for RawSugar data (http://www.rawsugar.com), some seem to group truly related tags (e.g., {`health, nutrition, food, diet`}), while tags in other clusters are less related (e.g., {`health, shopping, research`}). No assumption can be made about the nature of the relationships holding within a cluster.

Wu et al. (2006) present a probabilistic model, which aims to generate groups of semantically related tags based on the co-occurrence of tags, resources, and users. Entities (user, resource or tag) are represented as a multi-dimensional vector, a *conceptual space***,** where each dimension represents a category of knowledge – whose meaning is unknown. The value in each dimension should measure the level of relationship between the entity and the corresponding category of knowledge. The log-likelihood of the dataset is estimated in order to determine the number of dimensions of that conceptual space and assign the relationship values of entities to each dimension. In experiments with a subset of del.icio.us data, 40-dimensions are estimated as sufficient to represent the major category of meanings. A small example taking 10 randomly selected dimensions and the top 5 closely related tags to each of those dimensions shows that relationships hold amongst the tags within each group (dimension), however, once more, the types of these relationships are not explored.

Mika (2005) extends the traditional bipartite model of ontology with a social dimension, yielding in a tripartite model involving users (actors), tags (concepts), and resources (instances of concepts). With a subset of del.icio.us' tags, based on the co-occurrence of tags with resources and users, the author builds graphs relating tags and users and also tags and resources. Techniques of network analysis, which are not discussed in the paper, are then applied on those graphs in order to discover emergent ontologies. For each graph, the result is a set of clusters of semantically related tags, but the relations are not made explicit.

Schmitz et al. (2006) extract association rules between projections of pairs of elements from the tripartite model of folksonomies, i.e., users, resources and tags. In experiments with data from del.ici.us, the authors illustrate two different projections, learning rules of the types: (i) users assigning certain tags to some resources often also assign another set of tags to those resources; and (ii) users labelling certain resources with a set of tags often also assign those tags to another set of resources. While these kinds of association rules make it possible to identify the existence of relationships among different tags, users or resources, they do not provide any information about the nature of these relationships.

Finally, most of the social systems, including Flickr and del.icio.us, provide facilities such as "clusters" and "related tags", which show groups of related tags to allow the user to tune the search to other (statistically) related tags. Del.icio.us also provides "recommended tags" and "popular tags" when a given resource is being tagged, based on tags previously used for the same resource. Apparently these facilities rely on co-occurrence information but the groups express nothing about the actual relationships between the tags.

Since none of the described approaches applies more sophisticated pre-processing of the tags than eliminating infrequent tags, tags like `Music` and `music` count as different elements. The same applies to tags with very little lexical variation, such as `blog` and `blogs`. As we describe in the next section, we use specific strategies for cleaning up tags, and, more importantly, besides identifying groups of related tags, we investigate the nature of these relationships by exploiting information available on the semantic web, in order to give semantics both to the tags themselves and to the relationships between tags.

3 Integrating Folksonomies with the Semantic Web

As we previously mentioned, the tag space in social tagging systems encompasses semantic aspects of the system that are not explicitly defined. By identifying formal elements corresponding to tags and relationships among them it is possible to make explicit a significant part of this underlying knowledge, which is crucial for the efficient use of these systems. In fact, only with a clear semantic structure the annotations in folksonomies can be useful not just to humans, but can be made available to software agents and applications on the semantic web. Our hypothesis is that this knowledge can be derived by means of a statistical analysis of the annotations combined with pragmatic information provided by the semantic web and additional clues given by external resources.

3.1 Datasets

We investigate the tag sets in Flickr and del.icio.us due both to their popularity (with a large number of resources, users, and tags) and availability. These datasets differ from each other in a series of features. In fact, Thomas Vander Wal[3] mentions these systems when distinguishing between broad and narrow folksonomies: in a broad folksonomy (e.g., del.icio.us) many users tag the same resource, while in a narrow folksonomy (e.g., Flickr) only the creator of the resource tags it. Other studies on the structure of both del.icio.us and Flickr, focusing on user activity, tag frequency, kinds and variability of tags, among other aspects, are presented in (Golder and Huberman, 2005) and (Marlow et al., 2006). In our experiments, we use the del.icio.us tags provided by Peter Mika, which were also used in (Mika, 2005), and Flickr tags for photos posted between 01-02-2004 and 01-03-2006. The total numbers of entries (i.e., a resource tagged by a user) and tags in both datasets, as well as the number of distinct users, resources and tags, are shown in Table 1.

3.2 Methodology

Two very important features of our methodology are that it is unsupervised, i.e., it does not assume previously identified mappings or relationships to train the system, and it does not require any context besides the tags themselves, the resources being tagged and other tags used for those resources. The general approach, as given in Fig. 1, consists of three steps: pre-processing, clustering and concept / relation identification.

[3] http://www.personalinfocloud.com/2005/02/explaining_and_.html

Table 1. Number of tags (with their corresponding users and resources) from del.icio.us and Flickr

	Total		**Distinct**		
	# entries	*# tags*	*# users*	*# resources*	*# tags*
del.icio.us	19,605	89,978	7,164	14,211	11,960
Flickr	49,087	167,130	6,140	49,087	17,956

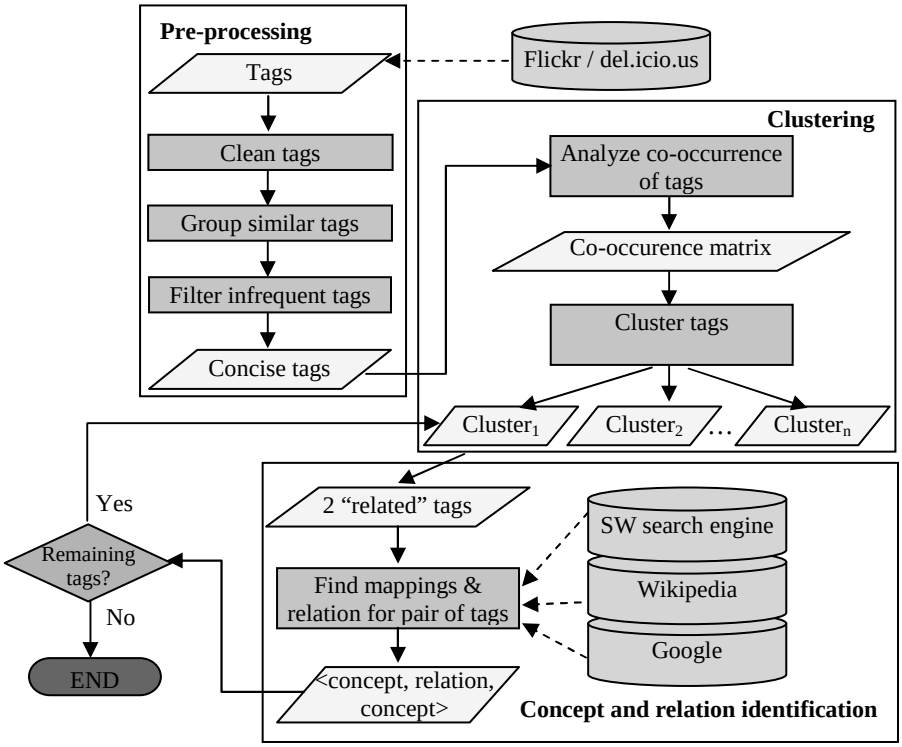

Fig. 1. System architecture

3.2.1 Pre-processing

The following shallow pre-processing steps were performed:

(1) Filter out unusual tags (and corresponding resources, if no other tag remains in that annotation). From a social perspective, all tags are relevant, even if they cannot be mapped to elements in ontologies. However, at this stage we are interested in tags with a more general applicability, which can be possibly found in ontologies, and therefore we define the following constraints: tags must start with a letter followed by any number of letters, numbers, and symbols like dash, dot, underscore, etc.

(2) Group morphologically very similar tags using the Levenshtein similarity metric[4] with a high threshold to determine "similar" words. This can tackle minor morphological variations (by grouping tags such as `cat` and `cats`,

[4]As implemented in the package SimMetrics in http://sourceforge.net/projects/simmetrics/

`san_francisco`, `sanfrancisco` and `san.francisco`) as well as misspellings (by grouping tags such as `theory` and `teory`). Within each group of similar tags, one is selected to be the representative of the group and all the occurrences of tags in that group are replaced by their representative. Given the alphabetically sorted list of tags within a group, the main criterion for choosing its representative is the existence of the tag in WordNet, followed by other simple criteria: preference is given to tags with letters only, followed by words with some symbols, then combinations of words and numbers, and so on. For example, the tags `typography`, `web-based`, and `tutor` were respectively chosen to represent the following groups:

```
{tipography typograph typography}
{web-based web_based webbased}
{tutor tutors}
```

(3) Filter out infrequent and isolated tags (and corresponding resource), that is, tags occurring less than a certain number of times or appearing only isolated.

3.2.2 Clustering

The second step of our approach is to perform a statistical analysis of the tag space in order to identify groups, or clusters, of possibly related tags. Clustering is based on the similarity among tags given by their co-occurrence. In order to find these similarities, the tags in each of the datasets were organized as a co-occurrence matrix, that is: a *n* x *n* symmetric matrix *M*, were *n* is the number of distinct tags in the dataset, and the value of each element m_{ij}, representing the intersection of tag_i and tag_j, corresponds to the number of times the pair tag_i and tag_j co-occur in the whole tagset (with the same or different resources/users). If $tag_i = tag_j$, then the intersection represents the frequency of the tag in the dataset.

In this co-occurrence matrix, each line (or column) is a vector representing one of the tags. Therefore, several vector space statistics can be computed. We tried different metrics to calculate the similarity between the pairs of vectors, including Euclidian and Manhattan distance, angular separation (cosine), etc. Metrics computing absolute distance like Euclidian and Manhattan showed to be inappropriate, since they are much more sensitive to significant variations in a few elements than little variations in a large number of elements, which is relevant to our problem. We chose angular separation, illustrated in (1), which computes the cosine angle between two vectors and thus is more sensitive to small changes in various elements. It is also less complex than similar metrics such as correlation coefficient.

$$angular\_separation_{ij} = \frac{\sum_{k=1}^{n} x_{ik}.x_{jk}}{\left(\sum_{k=1}^{n} x_{ik}^2.\sum_{k=1}^{n} x_{jk}^2\right)^{\frac{1}{2}}} \qquad (1)$$

As a result of computing the similarity between each possible pair of vectors in the co-occurrence matrix (i.e., *n* x *n* pairs), for each tag we obtain a list of its similarities to all the other tags. For example, Table 2 shows the top five similar words to the words `audio`, `semantic-web`, `adult`, `apple`, and `chat` in del.icio.us data.

As we can see in Table 2, this co-occurrence-based similarity computation already shows some semantics about the words. It goes beyond finding syntagmatic associations, since we

do not simply check the pairs of tags that co-occur a significant number of times, such as in (Begelman, 2006). In our case, by using the co-occurrence matrix, we take into account all the other tags as context and state that to be considered similar to a certain tag_j, a tag_i has to co-occur not only with tag_j but also with the other tags co-occurring with tag_j. That is, both tags must have a similar pattern of co-occurrence, which is given by their co-occurrence vectors. Similarities like these are sometimes called "paradigmatic associations".

Table 2. Top 5 similar words to some examples of tags

Top	**audio**	**semantic -web**	**adult**	**apple**	**chat**
1	mp3	rdf	girls	mac	aim
2	music	ontology	nude	macintosh	messenger
3	playlist	owl	babes	tiger	gtalk
4	streaming	semweb	pics	osx	msn
5	radio	daml	sex	macosx	icq

Certainly, a single tag can have two or more patterns of co-occurrence, representing different meanings or uses of the tag (e.g., `apple` as *computer brand* and as *fruit*). In that case, the most similar tags to a given tag will mix words referring to distinct domains. Therefore, although relevant, the information provided by the paradigmatic associations is limited to pairs of tags, i.e., it can only tell, for a given tag, that there is a set of other tags that are related to it, but this does not guarantee that a relationship also holds among the other tags in that set. For example, we could also have found a paradigmatic association between **`apple`** and **`fruit`** in Table 2, but clearly we should not include `fruit` in a group representing the *computer brand* sense of `apple`. Therefore, we extend the paradigmatic associations by defining a clustering algorithm on top of them.

In order to group the highly co-occurring tags, we first establish a similarity threshold to filter out pairs of tags that are not highly similar. Given the highly similar pairs of tags, the algorithm takes into account the mutual similarity amongst tags to identify the groups. It considers each pair of similar tags, for example, `audio` and `mp3`, as seeds constituting an initial cluster, and then tries to enlarge this cluster by looking for tags that are similar to both the initial tags. This procedure is recursively repeated for all the tags, i.e., each new "candidate" tag for a cluster must be similar to the whole (possibly enlarged) set of tags in that cluster. Once there are no more candidates for that cluster, a new pair of similar tags (e.g., `audio` and `music`) is taken as seed and this is repeated until all pairs of tags have been processed[5].

This procedure generates a set of clusters, including a number of identical clusters, resulting from distinct seeds that are in fact similar amongst each other. It also generates highly similar clusters, differing in only a few tags, which are in many cases a consequence of the threshold to filter out not so similar pairs of tags. We use two smoothing heuristics to avoid having a high number of these very similar clusters. For every two clusters:

[5] Our clustering strategy can be compared to the *Clustering by Committee* approach (Pantel, 2003), in the sense that multiple elements are chosen to be the cluster's centroids, as opposite to traditional partitional approaches in which only one centroid per cluster. However, our strategy is more strict, since all the elements within a cluster must be similar amongst each other, instead of being similar just to the centroids.

1) If one cluster contains the other, that is, if the larger cluster contains all the tags of the smaller, remove the smaller cluster;
2) If clusters differ within a small margin, that is, the number of different tags in the smaller cluster represents less than a percentage of the number of tags in the smaller and larger clusters, add the distinct words from the smaller to the larger cluster and remove the smaller.

These heuristics make it possible to group two tags that are not sufficiently similar according to the established threshold, but are both similar to a large set of other tags. Therefore, we are able to eliminate redundancies but keep multiple clusters sharing a number of tags when those tags have multiple meanings, indicating that they are ambiguous tags. Good quality resulting clusters can already be used for several of the tasks described in Section 1, including tag extension/disambiguation, visualization and suggestion.

One important feature of our clustering technique is that it does not require establishing the number of clusters to be produced. The only parameters are the threshold to define the minimum co-occurrence for pairs of tags and the percentage of variation allowed for "similar clusters". Alternatively, we could have used traditional clustering algorithms. However, as we discuss in Section 5, this approach showed to be more appropriate for our problem.

3.2.3 Concept and Relation Identification

Since our clusters are derived from co-occurrence information only, there is no indication of the relationships holding amongst subsets of the tags in each of them. Our goal is to use knowledge provided by different sources, including ontologies available on the semantic web, Wikipedia and Google, to discover if there are in fact relationships between tags in each cluster and, if they exist, categorize them. This process involves mapping the tags into concepts / instances / properties of ontologies and checking the possible relationships among the mapped tags. As a source for ontologies we use semantic web search engines such as Swoogle (Ding et al., 2004). The procedure within each cluster is the following:

1) Post each possible pair of tags to the semantic web search engine in order to retrieve ontologies that contain both tags. All combinations of pairs are tried, since it is not possible to know within which pairs a relation holds (look for matches with labels and identifiers).

2) If any of the tags is not found by the search engine, consider that they can be acronyms, misspellings or variations of known terms, and look for them in additional resources:

2.1) Post the tag to Wikipedia in order to get (in the title field) the whole expression in case it is an acronym. For example, the query term `NYC` in Wikipedia returns as title `New York City`. Select the text in the title.

2.2) If the tag is not in Wikipedia, consider it to be a misspelling or multi-word term. Post it to Google, looking for a suggestion of correct term. For example, in a search for `sanfrancisco` in Google, the system returns the following suggestion: "`Did you mean: san francisco?`" Select the suggested term.

3) If the two tags (or the corresponding terms selected from Wikipedia or Google) are not found together by the semantic web search engine, consider them not to be related and eliminate the pair from that cluster if they are not (possibly) related to any other tags, that is, all the combinations of pairs of tags must be searched.

4) Conversely, if ontologies are found containing the two tags:

4.1) Check whether the tags were correctly mapped into elements of the ontologies. Tags can refer to the following elements: concepts, instances, or properties.

4.2) Retrieve information about the tags in each of the ontologies: the type of tag (concept, instance, property), its parents (up to 3 levels) if it is a concept or an instance, and its domain and range or value if it is a property.

5) For each pair of tags for which the semantic web search engine retrieved information, investigate possible relationships between them:

5.1) A tag is an ancestor of the other. For example, in the *Food* ontology[6], `apple` is a subclass of `fruit`.

5.2) A tag is the range or the value of one of the properties of the other tag. For example, in the *Wine* ontology[7], the class representing the wine `Zinfandel` has a property `hasColor`, for which the value is `red`. Therefore, the relation `hasColor` holds between `Zinfandel` and `red`. This extends to properties defined in superclasses of the actual classes of the tags.

5.3) Both tags have the ***same direct parent***. For example, `apple` and `pear` are concepts with the same parent (`fruit`) in the `FOOD` ontology.

5.4) Both tags have the ***same ancestors***, at the ***same level***. For example, in WordNet (Miller et al., 1994), `assembly` has as ancestors `building` (1st level) and `construction` (2nd level), while `formation` has the ancestors `fabrication` (1st level) and `construction` (2nd level).

5.5) Both tags have the ***same ancestors***, at ***different levels***. For example, in WordNet, `chapterhouse` has as ancestors `building` (1st level) and `construction` (2nd level), while `edifice` has as ancestor `construction` (1st level).

By looking for pairs of tags in single ontologies, instead of individual tags, we eliminate much of the ambiguity in those tags. For example, in the case (5.3) above, `apple` is also defined in other ontologies with different meanings, for example, the *computer brand* in the CLib-core-office ontology[8]. However, this ontology does not contain the tag `pear`, and thus it is not considered an information provider for the relation identification.

If more than one ontology contains both tags (and possibly relationships between those tags), we currently use a simple resolution strategy: we give preference to the ontology containing also other tags (and possibly relationships) in the cluster. If there are still multiple ontologies, the first fulfilling this constraint is chosen. If there are multiple relationships between a pair of tags in that ontology, we choose the first of them according to steps (5.1-5.5).

If the aforementioned procedure does not yield any relationship for a given pair of tags co-occurring in at least one single ontology, we assume they are not related and remove the pair from the current cluster (unless they are related to other tags). Obviously these strategies are rather simplistic and in the future we plan to use more elaborate strategies, in particular for deciding whether to merge information coming from multiple ontologies and how to do so (Sabou et al., 2006). It is important to notice that even when relationships are not found, if the tags can be correctly mapped to elements in ontologies, these mappings can already be

[6] http://lists.w3.org/Archives/Public/www-archive/2004Oct/att-0016/food.owl
[7] http://www.w3.org/TR/2003/CR-owl-guide-20030818/wine
[8] http://www.cs.utexas.edu/users/mfkb/RKF/tree/CLib-core-office.owl

used to support the various tasks discussed in Section 1. For example, related elements (concepts, instances, and properties) in the ontology, like subclasses of a concept representing a tag, can be used to extend searches. However, going one step ahead and finding self-contained structures within each cluster by merging knowledge from multiple ontologies can provide a much richer perspective on the underlying semantics of the tagging systems. In what follows we present initial experiments with our approach to find both meaningful clusters and the underlying relationships amongst their tags.

4 Experiments and Discussion

For both del.icio.us and Flickr datasets, in the first step of the approach, i.e., the **pre-processing** strategies, we empirically defined the parameters as follows: (1) to be "similar", a pair of words has to reach **0.83** or higher score (Levenshtein metric); (2) a tag has to occur at least **10** times. The resulting number of tags, resources, and users is shown in Table 3.

Table 3. Number of tags (and their corresponding users and resources) after the pre-processing steps

	Total		**Distinct**		
	# entries	*# tags*	*# users*	*# resources*	*# tags*
del.icio.us	18,882	70,194	7,090	13,579	1,265
Flickr	44,032	127,098	5,321	44,032	2,696

For the **clustering** step, after generating the co-occurrence matrix for each of the datasets and computing the similarity between all pairs of vectors (tags) in that matrix, we empirically established a similarity threshold of **0.5** for both datasets to filter out the pairs of tags that were not highly similar. This means eliminating a number of tags that are not similar enough to any other tag. Excluding symmetric pairs, this resulted in **2,298** pairs of tags (**847** distinct) in del.icio.us, and **4,983** (2140 distinct) in Flickr.

For the smoothing heuristics to avoid a high number of very similar clusters, we established the threshold of **0.3** as accepted difference to group two "similar" clusters, that is, the number of distinct tags cannot be greater than **30%** of the number of tags in each of the clusters. This resulted in **410** clusters for del.icio.us, and **882** for Flickr.

The high number of clusters is due to the existence of clusters with only two tags. This can mean simply that certain pairs of tags do not co-occur with other tags a significant number of times. However, it may also indicate that the tags in the pair constitute a compound word, co-occurring between each other much more than with any other tag. This happens mostly in Flickr data, where we found clusters containing pairs of words such as `{el salvador}`. If we discard clusters with two tags, the number of clusters decreases to **47** in del.icio.us and **206** in Flickr. Some examples of clusters are shown in Table 4.

Alternatively to our clustering strategy, we experimented with a traditional clustering algorithm, namely, k-means, using cosine as the similarity metric. K-means requires the number of clusters to be given a priori, and therefore we defined the following numbers: 30, 50, 100, 150 and 200. Amongst the generated clusters, some seem to be very good, but many of them contain noise and some are meaningless. One of the reasons for that is the need of defining the number of clusters, which can force clusters to be divided when it is not necessary (or vice-versa). Also, it is not possible to discard pairs of tags that do not co-occur a certain number of times and therefore all the tags will be in the resulting clusters. A final

problem is that the random criterion to create seed clusters can yield completely different clusters at each run. In future we will experiment with a hybrid strategy involving hierarchical clustering, which does not require defining the number of clusters, and a means of establishing a similarity threshold to discard tags.

Table 4. Examples of clusters found for del.icio.us and Flickr data

del.icio.us data
{author books literature}
{bicycle bike courier cycling}
{bingo blackjack casino gamble gambling keno poker roulette slots}
{bookmark bookmarking folksonomy social tagging tags}
{browse extension firefox mozilla thunderbird}
{aim chat gtalk icq instant jabber messaging messenger msn yahoo}
Flickr data
{activism anarchism banner brutality demonstration eu globalization gothenburg police protest riots summit syndicalism worker}
{backpacking hot humid iguacu lush rainforest waterfall wilderness}
{damage flooding hurricane katrina Louisiana}
{bosnia europe herzegovina Sarajevo}
{apple ibook mac ipod macintosh powerbook}

By looking at the clusters obtained for both tag sets, we found that clusters from del.icio.us express concepts that are apparently closer to formal categories than clusters from Flickr. This may be due to the distinct purposes of the two systems. In fact, more pre-processing steps seem to be necessary to allow identifying meaningful categories in Flickr. For example, dates in various formats could be mapped into a semantic category "date", while names of unknown people could be mapped to a category "person".

As we discussed in Section 3, meaningful clusters can be very useful to enhance certain tasks in folksonomies. However, systematically evaluating the quality of clusters is a very complex task, since it relies on subjective criteria. One way of carrying out this evaluation could be verifying whether the tags within each cluster are correctly mapped into elements in ontologies, as we show with a few examples in what follows while describing the next step of our experiments.

For the last step of the approach, i.e., the **concept** and **relation identification**, we used Swoogle, since it is the most comprehensive semantic web search engine available on the web right now. Although Swoogle can provide useful information, it does not take into account semantic particularities of the data when creating indexes, and this yields severe limitations for our purposes. Querying facilities are limited to keyword search, with a few modifiers allowed, while we need more fine-grained queries, which would allow distinguishing between concepts, instances, properties, etc. In the presentation of the results, only part of the information is returned in a structured way and thus in most of the cases it is necessary to download and parse the ontologies. In the near future, we will use a semantic web search engine under development in our group, Watson (d'Aquin et al., 2007), which aims to overcome these limitations.

Given the difficulties to find the information we need in Swoogle, we performed a few experiments considering the search for *concepts* only, with *exact matching*. Tags within a cluster were queried using the *Ontology Dictionary* facility in Swoogle 2005 and the retrieved ontologies were manually analyzed to find both mappings to concepts and

relationships. The examples illustrated here aim to show the potential of the approach in finding mappings and relationships: a fully automated version will be implemented when our semantic web search engine is ready.

Starting from the previously obtained clusters (410 for del.icio.us: and 882 for Flickr), we posted all the possible pairs of tags within each cluster to Swoogle. Out of a total of **3152** pairs (**847** distinct tags) in del.icio.us and **5031** pairs (**2140** distinct tags) in Flickr, without using Wikipedia / Google to find unknown concepts in Swoogle, the following numbers of pairs were found as concepts together in at least one single ontology: **569** pairs (**358** distinct tags) in del.icio.us and **309** pairs (**492** distinct tags) in Flickr.

Many of the pairs were found in WordNet only. WordNet, which is more like a dictionary than an ontology, and thus has a wide coverage over most of the concepts. In general, finding two concepts in any ontology does not mean that they are related, but this is even more critical with WordNet: concepts are more likely to be related via very generic semantic categories, such as "entity" or "thing". Moreover, WordNet is very limited in terms the relationships that are covered: mostly hierarchical. If we consider only the pairs of tags that were found in other ontologies than WordNet, the numbers decrease to **126** (**97** distinct tags) in del.icio.us and **67** (**94** distinct tags) in Flickr. This means that **97** tags in del.icio.us and **94** in Flickr could be mapped to concepts in ontologies (except WordNet). In order to assess the quality of these mappings, and therefore the quality of the clusters containing the corresponding tags, we manually verified whether the concepts identified in ontologies were in accordance with the knowledge represented by their cluster. For example, in *Cluster_1* in Fig. 2, all the tags except `sourcecode` where found in ontologies, i.e., there was a possible mapping for all the other tags, even if no relationship was found for many of them. The tags for which a valid relationship was found are considered correctly mapped. Out of the remaining 7 tags, namely `collection control dom form layout program repository`, only `dom` could not be correctly mapped to a concept in any ontology. We believe this extends to most of the clusters: there will not be a valid mapping for only a few tags in clusters. This may indicate that these tags do not belong to any of the clusters, but also may just reflect the low coverage of the current search engine over the ontologies available on the web. However, it is important to notice that the most frequent tags in the two datasets, which are supposedly some of the most relevant, are amongst the pairs of tags that were found in ontologies.

Regarding the identification of relationships, it is important to remember that not all the pairs of tags in a cluster are expected to be related. In fact, the most common situation is that a certain tag is related to a few others, which are, in turn, related to others, composing an incomplete directed acyclic or cyclic (possibly disconnected) graph. Without considering the pairs found in WordNet, in Fig. 2 we show examples of partial ontologies that were produced for some del.icio.us clusters by following the steps 5.1-5.5 (Section 3). Terms in italic are not part of the cluster, but were kept in the graphs to indicate indirect relationships. Arrows without explicit relationships represent "**subclass**" relations.

As we can see, not all the tags in each cluster are included in the graphs. This may be because either the tag was not found in Swoogle at all (e.g., `lms` in *Cluster_2*), the tag was not found in a single ontology together with at least one of the other tags (e.g., `sourcecode` in *Cluster_1*), or no relationship was found within an ontology (e.g., `layout` in *Cluster_1*). When a tag is not found in Swoogle, additional resources can be used. In *Cluster_3* these additional resources play a very important whole. Swoogle does not contain concepts referring to `distro`, `fedora`, `gentoo`, `kubuntu`, `mandriva`, or `rpm`. The only

relationships that could be retrieved were: `suse` and `debian` are subclasses of `linux`. By using Wikipedia, we could infer the relationships between most of the other tags and `linux`.

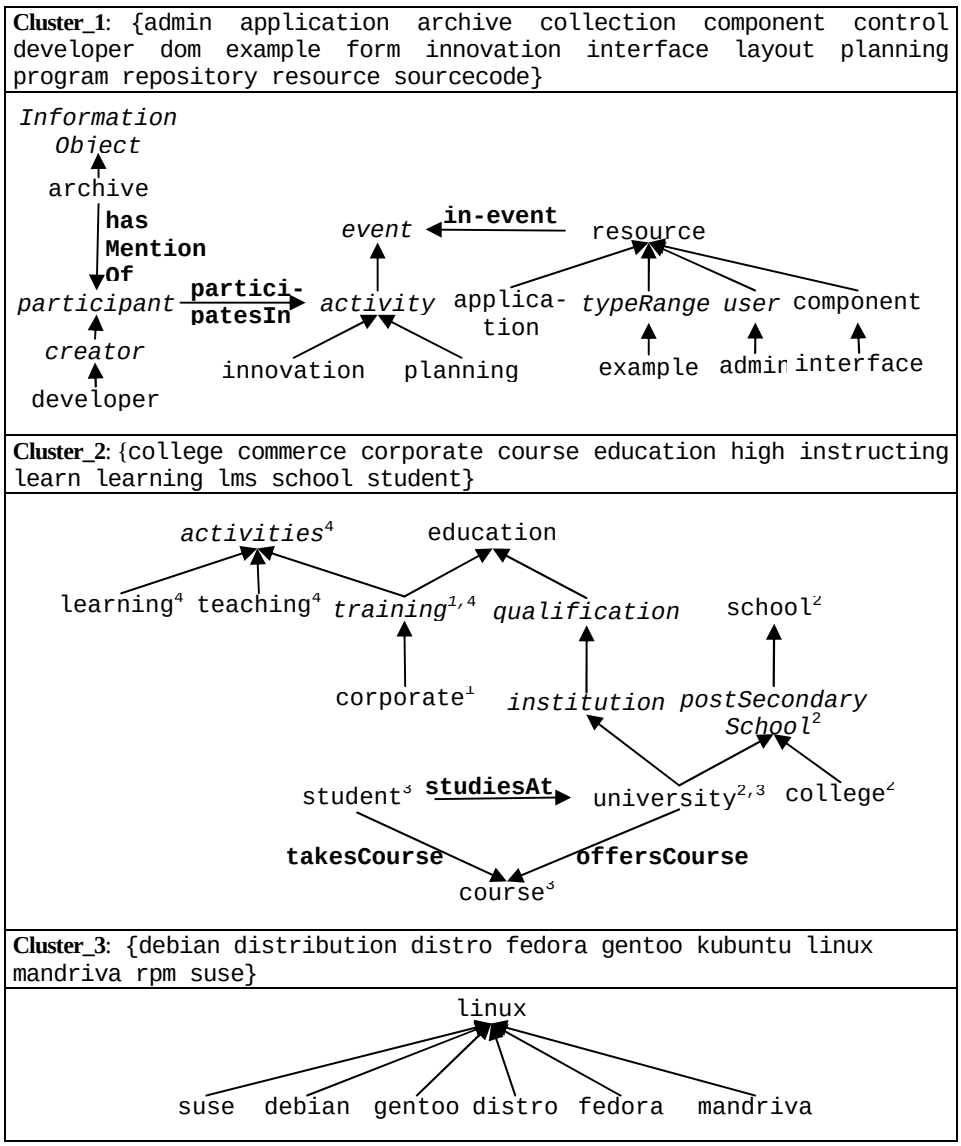

Fig. 2. Examples of relationships for del.icio.us clusters

5 Conclusions and Future Work

We have presented some initial work in the direction of integrating folksonomies with the semantic web, thus making explicit the semantics behind the tag space in folksonomies.

Preliminary experiments with tags from del.icio.us and Flickr have shown that the approach is feasible and very promising: meaningful groups of tags corresponding to concepts in ontologies could be derived by means of co-occurrence analysis and clustering techniques, while relationships within tags in each cluster could be discovered by querying ontologies in Swoogle.

In the future we plan to improve the approach in several aspects. These include: (i) using a new clustering technique which combines hierarchical clustering with a threshold to discard tags that are not sufficiently similar to others, and (ii) implementing a fully automated version of the last step of the approach, that is, the process to map tags into ontology elements to build partial structures based on the knowledge provided by our new semantic web search engine. In order to achieve this goal we will also need to devise better strategies for ontology selection and matching, as well as strategies to extract information from external resources (Wikipedia / Google). We also plan to extrinsically assess the quality of our results by integrating them in the context of the various tasks discussed in the paper (tag disambiguation, result visualization, ontology evolution, etc).

References

Aurnhammer, M., Hanappe, P., Steels, L.: Augmenting Navigation for Collaborative Tagging with Emergent Semantics. 5th ISWC, Athens, GA, LNCS 4273 (2006) 58-71.

Begelman, G., Keller, P., and Smadja, F.: Automated Tag Clustering: Improving search and exploration in the tag space. Collaborative Web Tagging Workshop, 15th WWW Conference, Edinburgh (2006)

d'Aquin, M., Sabou, M., Dzbor, M., Baldassarre, C., Gridinoc, L., Angeletou, S. and Mottta, E.: WATSON: A Gateway for the Semantic Web. Poster Session at 4th ESWC (2007)

Ding, L., Finin, T., Joshi, A., Pan, R., Scott Cost, R., Peng, Y., Reddivari, P., Doshi, V.C., and Sachs, J.: Swoogle: A Search and Metadata Engine for the Semantic Web. 13th ACM Conference on Information and Knowledge Management, Washington D.C. (2004)

Golder, S., and Huberman, B.A.: The Structure of Collaborative Tagging Systems. HP Labs technical report. (available in http://www.hpl.hp.com/research/idl/papers/tags/) (2005)

Marlow, C., Naaman, M., Boyd, D., Davis, M.: Position Paper, Tagging, Taxonomy, Flickr, Article, ToRead. Collaborative Web Tagging Workshop, 15th WWW Conference, Edinburgh (2006)

Mika, P.: Ontologies are us: A unified model of social networks and semantics. 4th ISWC (2005)

Miller, A., Chorodow, M., Landes, S., Leacock, C., Thomas, R.G.: Using a Semantic Concordancer for Sense Identification. Arpa Human Language Technology Workshop (1994) 240-243

Motta, E and Sabou, M.: Next Generation Semantic Web Applications. In Mizoguchi et al. (eds.), The Semantic Web - ASWC 2006, LCNS 4185, Springer (2006)

Pantel, P. Clustering by Committee. Ph.D. Dissertation. University of Alberta (2003)

Sabou, M., d'Aquin, M., Motta, E.: Using the Semantic Web as Background Knowledge for Ontology Mapping. International Workshop on Ontology Matching, Athens, GA (2006)

Schmitz, C., Hotho, A., Jäschke, R. Stumme, G.: Mining Association Rules in Folksonomie. IFCS Conference, Ljubljana (2006) 261-270

Schmitz, P.: Inducing ontology from Flickr tags. Collaborative Web Tagging Workshop, 15th WWW Conference, Edinburgh (2006)

Schutz, A. and Buitelaar, P.: RelExt: A Tool for Relation Extraction from Text in Ontology Extension. 4th ISWC, Galway (2005) 593-606

Specia, L., Motta, E.: A hybrid approach for extracting semantic relations from texts. 2nd Workshop on Ontology Learning and Population at COLING/ACL 2006, Sydney (2006) 57-64

Wu, X., Zhang, L., Yu, Y.: Exploring social annotations for the semantic web. 15th WWW Conference, Edinburgh (2006) 417-426