Using Social Media for Ontology Enrichment

Paola Monachesi and Thomas Markus

Utrecht University, Utrecht, The Netherlands

Abstract. In order to support informal learning, we complement the formal knowledge represented by ontologies developed by domain experts with the informal knowledge emerging from social tagging. To this end, we have developed an ontology enrichment pipeline that can automatically enrich a domain ontology using: data extracted by a crawler from social media applications, similarity measures, the DBpedia knowledge base, a disambiguation algorithm and several heuristics. The main goal is to provide dynamic and personalized domain ontologies that include the knowledge of the community of users.

1 Introduction

Social media applications are accessed by millions of users that actively participate in the creation of textual and visual content, providing tags to describe the resources they have contributed. Social media begin to acquire relevance also in educational contexts with learners relying on them for learning purposes. For example, the Massachusetts Institute of Technology has a channel for posting videos on Youtube that has 57,000 subscribers and the channel page has been viewed for more than 1 million times. On YouTube, there are videos of lectures given in top universities that have more than 50,000 views.

There is thus the need to support the learners in accessing and exploiting this material in the most appropriate way. One possibility is to employ the tags provided by users to the resources in order to provide search results and recommendations about the most relevant material for the given learning task. However, [13] reports that learners don't find tag clouds particularly useful when searching for learning material since they only show relations among topics but they don't provide information about how the topics are related. Ontologies prove to be a more valuable support than tag clouds in the knowledge discovery process. They provide a clear structure which is based on relations among concepts that can be also very useful in discovering new topics and associations. More specifically, domain ontologies can guide and support the learner in the learning path, facilitate (multilingual) retrieval and reuse of content as well as mediate access to various sources of knowledge, as concluded in [14]. However, this formalization might not always correspond to the representation of the domain knowledge available to the learner which might be more easily expressed by the tagging emerging from communities of peers via available social media applications.

L. Aroyo et al. (Eds.): ESWC 2010, Part II, LNCS 6089, pp. 166–180, 2010.

In the context of the *Language Technology for LifeLong Learning* project,[1] we propose an ontology enrichment methodology that complements the formal knowledge represented by domain ontologies with the informal knowledge emerging from tagging. More specifically, in our approach, we include the expert view on the domain by maintaining the ontology structure but we complement it with the 'wisdom of the crowd' emerging from tagging. We provide thus more dynamic ontologies that take into account the evolving vocabulary of the Community of Practice. Similarity measures have been evaluated and are employed to identify tags which can be related to the concepts of an existing domain ontology. A knowledge base such as DBpedia [1] is used in order to map the tags into the ontology in combination with a disambiguation mechanism. However, tags are also related to users providing information about their interests, their knowledge and their level of expertise within a domain. The MOAT ontology is employed to create a link between users and the meaning of various tags, allowing thus the identification of the vocabulary of Communities of Practice.

The paper is organized as follows. Section 2 discusses the state of the art. Section 3 introduces the ontology enrichment process and its various components. Section 4 discusses the role that similarity measures can play in enhancing ontologies with tags while section 5 focuses on reference knowledge bases such as DBpedia to map the relevant tags into an existing ontology. Section 6 presents the approach to tag disambiguation adopted and the methodology employed to create ontologies related to Communities of Practice. In section 7, we evaluate the resulting ontology. Finally, in section 8, we discuss some future work and perspectives.

2 State of the Art

There are two main approaches to structure the tags extracted from social media applications in order to organize them and to understand their meaning. The former approach relies on the information that can be retrieved from the social media applications such as users, tags and tagged resources. In particular, [17] and [8] developed algorithms to derive a hierarchy of tags, based on the data of a tripartite tagging network. Three measures for relatedness are compared in [4]: one measure is based on co-occurrence, one is based on the cosine similarity, and then there is the FolkRank algorithm. The measures are applied to find a set of closely related tags. Subsequently, the authors propose a mechanism of semantic grounding: the found tags are mapped to WordNet, in order to inspect the semantic distance between the related tags. Cosine similarity appears to yield more synonyms, where the other two measures rather yield different concepts, among which are superconcepts, which make them appropriate for retrieving taxonomic relationships. FolkRank, in addition, is capable of detecting multi-word lexemes from distinct tags. In [20], the tag-resource-user relations are represented as multidimensional vectors and they use a probabilistic model to find categories of knowledge.

[1] http://www.ltfll-project.org/

The latter approach exploits external semantic resources to structure sets of tags. An example is provided by [19] that tries to make explicit the semantic structure of tagging for semantic web applications. They describe an approach using tag preprocessing (morphologic similarity, exclusion of isolated tags), statistical tag clustering based on co-occurrence, and relation identification by looking up terms in online ontologies. A similar approach is described in [6] which, however, focuses on how an actual ontology can be generated on the basis of a folksonomy. They propose that in addition to providing the tags, the community can also directly help to identify or judge/approve relations in the ontology.

The work presented in this paper relies on the techniques previously described and integrates them in a comprehensive and automatic approach to ontology enrichment in which similarity measures are combined with external semantic resources. In addition, while the approaches mentioned above are an attempt to develop (light) ontologies from sets of tags, our proposal differs from them because it relies on existing domain ontologies. More specifically, we embed the tags extracted from social media application into the structure of an existing ontology. It is thus possible to exploit the growing number of ontologies available as result of the Semantic Web initiative and enhance them with the extended vocabulary of Community of Practices arising from social data. This is a more relevant use of existing ontologies than that suggested in [19], in which ontologies are exploited to discover possible relations among tags. Existing ontologies are normally limited in size and in domain, making it quite difficult to find enough relations. We believe that a large background knowledge base such as DBpedia [1] is much more appropriate for the relation discovery task.

3 Ontology Enrichment with Social Data

Domain specific ontologies are relevant for learners, since they offer structured information on a certain topic which is lacking in large background semantic resources, such as DBpedia.

Domain ontologies, however, might be too static since it is quite demanding to update them on a regular basis. They usually represent the conceptualization of a domain by experts. However, the conceptualization and the vocabulary employed by the expert might differ substantially from that available to the learner. We have developed a methodology that allows for the enrichment of an existing ontology on the basis of the vocabulary of the Community of Practice that the learner is part of. More specifically, the resulting ontology integrates socially relevant concepts within the structure of an expert view domain ontology. This makes the enriched domain ontology more accessible for use by a variety of learners.

In [13], experimental evidence is provided to show that an ontology enriched with social tags constitutes a useful approach in the context of a learning task. Both beginners and advanced learners agreed that it can be a valuable tool in knowledge discovery tasks since it can provide structure to the heterogeneous list of documents which constitutes the output of search engines.

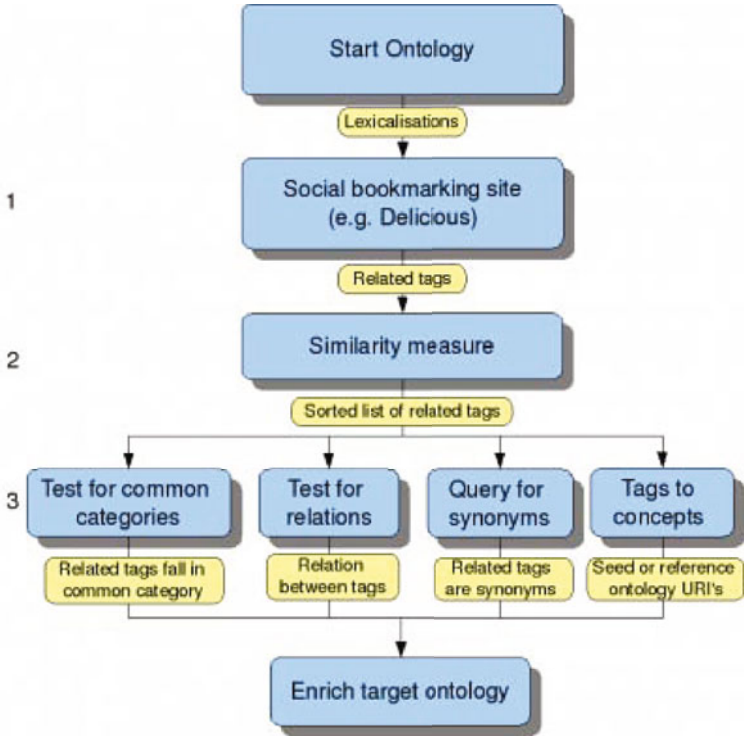

Fig. 1. The ontology enrichment pipeline

We have developed an ontology enrichment pipeline that can automatically enrich a domain ontology using data extracted by a crawler from social media applications, similarity measures, the DBpedia knowledge base, a disambiguation algorithm and several heuristics, as illustrated in figure 1.

The figure shows the steps involved in the enrichment process. More specifically, we have taken as starting point the LT4eL domain ontology on computing that was developed in the Language Technology for eLearning project.[2] It contains 1002 domain concepts, 169 concepts from OntoWordNet and 105 concepts from DOLCE Ultralite. The connection between tags and concepts is established by means of language-specic lexicons, where each lexicon specifies one or more lexicalizations for each concept [10].

As first step in the process, we extract data by means of a crawler that uses APIs provided by the social networking applications to get information about users, resources and tags. The crawler extracts links to resources from social media applications such as Delicious, YouTube and Slideshare together with the tags used to classify the resources and information about the social connections developed inside these web sites. The data extracted by the crawler can be interpreted as a folksonomy, which is a hypergraph describing the information

[2] http://www.lt4el.eu

about users, resources and tags, as specified in [11]. In the second step of the enrichment process, similarity measures are employed to identify tags that are related to the lexicalization of concepts already existing in the LT4eL domain ontology (cf. section 4). In the last step, we attempt to identify relations among the existing LT4eL concepts and new concepts derived from the tags, by relying on a background ontology such as DBpedia. Several heuristics are employed to discover taxonomic relations, synonyms and new relations explicitly coded in DBpedia (cf. section 5). Ambiguities are resolved through an appropriate disambiguation algorithm (cf. section 6).

4 Similarity Measures

Various similarity measures have been investigated in order to assess their possible contribution to automatic ontology enrichment. More specifically, our goal was to test which measures would allow for identification of more specific terms, alternative lexicalizations of pre-existing concepts and whether it would be possible to identify the relevant domain in case of ambiguity.

The similarity measures were applied to a large Delicious dataset which was previously aggregated with the crawler described in the previous section. It contains 598379 resources, 154476 users and 221796 tags on a wide range of subjects, but with an emphasis on computer related terminology.

In the rest of this section, we describe the various algorithms that were implemented as web services in order to assess their possible application in the ontology enrichment process.

Co-occurrence. In order to implement co-occurence, a tag-tag co-occurrence graph is calculated. This is a weighted, undirected graph. Two tags are connected if there is at least one post containing both tags. The weight of an edge is given by the number of posts that contain both t1 and t2. Given a tag t, all tags t2 such that weight (t,t2) is maximal gives a set of most related tags. This measure provides valuable input to extract taxonomic relationships between tags, as indicated by [4]. The co-occurrence measurement is also used by [18]. However, instead of using the measurement as described above, they point out that this measure should be normalized. There are two normalization methods:

- Assymmetric
- Symmetric: According to the Jaccard coefficient

The notion of co-ocurrence has been further developed into resource co-ocurrence and user co-ocurrence. In user co-ocurrence, the individual users are taken into account when calculating the co-ocurrence scores. A tag only co-occurs with another tag if that specific user actually added the two tags. This is the type of coocurrence that is defined by [4]. This is different from resource co-ocurrence where tags are said to co-occur when added to the same resource (by different users).

Cosine similarity. Given two vectors, cosine similarity is used to compute the similarity between two tags. The vectors can be computed in different ways, which leads to the distinction of different approaches.

- Tag Context Similarity: For each tag, a vector is created with as length the number of tags. The weights are the co-occurrence values of two tags. [4] points out that this method is suitable for finding synonyms and [19] used this measure to cluster tags.
- Resource Context Similarity: For each tag, a vector is constructed with as length the number of resources. The number of times the tag is used to annotate a resource determines the weight (tf). In [4], it is showed that this method is also suitable for finding synonyms. In [7], tf * idf is used in addition to the original method (tf). In all cases, they found tf * idf to be superior. They tried three different methods using this measure to cluster tags: hierarchical clustering, maximal complete link clustering and k-means clustering. The latter one yielded poor results, whereas hierarchical clustering had the best performance.
- User Context Similarity: For each tag, a vector is constructed with as length the number of users. Number of times the tag is used by a specific user, determines the weight.
- Document-Term similarity: This method is used in [3] that has applied this technique on Technorati Data. Similarity is calculated here from textual similarity of documents they annotate. This method is therefore not useful when tags are applied to images, videos etc. They induced a hierarchy of tags using similarity of the articles that were tagged.

4.1 Evaluation of Similarity Measures

We have evaluated the various similarity measures in the context of the ontology enrichment process and we have taken into account how many users and resources are necessary to obtain appropriate results. We have created a standard set of evaluation tags for which we have verified that our aggregated dataset contains enough information. This set contains 12 terms within the computing domain with different levels of abstractness. Since some measures can return thousands of results which would take too long to be evaluated manually, the analysis focused on the first 20 items. This means that for each of the measures 12*20=240 results were analyzed. The 12 standard test terms were: java, docbook, xml, xhtml, css, tex, standards, linux, design, blog, tools, software.

All the coocurrence measures were applied to this standard list and their results were analyzed. Two domain experts evaluated whether the output from the similarity measures consistently matched one or more of the possible output criteria. The different similarity measures were rated on a 5-point scale for each of the following criteria:

- Does the measure return concepts which are similar to the input term (e.g. java and jre)?

Similarity method	Similar concepts	Synonyms	Tail useable	Close in hierarchy
Resource Coocurrence (Jaccard)	5	1	1	4
Resource Coocurrence (Assymetric)	3	1	1	1
User Coocurrence (Jaccard)	5	1	1	5
User Coocurrence (Assymmetric)	3	1	1	1
Resource Cosine Similarity	5	3	1	4
User Cosine Similarity	3	1	1	3

Fig. 2. Results of the evaluation of similarity measures

- Does it reliably list synonyms at the top of the result list (e.g. cpu and processor, javascript and ecmascript) ?
- Is it possible to find a pattern (spelling error, unrelated term) (html and cookies) within the results found in the tail (items with the lowest score)?
- Are the related tags close to each other in the ontological hierarchy, taking the existing LT4eL domain ontology as a point of reference (e.g. xhtml and xml)?

The table in fig. 2 gives an overview of how the similarity measures perform with respect to each of the criteria mentioned above.

It shows that the different normalization methods for co-ocurrence greatly influence the results returned. A detailed analysis of the results, indicates that the data could be very useful in a manual enrichment of the ontology. However, the results are less useful if the goal is an automatic ontology enrichment process, as in our case. For example, the first hits for asymmetric co-ocurrence are very generic which are of little value because the relation to the input term is too trivial.

User co-ocurrence was found to be roughly equivalent to resource co-ocurrence for larger number of resources and users [13]. The results suggest that a small number of users doesn't need to be a problem with respect to the representativeness of the result as long as enough resources are tagged. If the user co-ocurrence similarity measure is employed, a sample of about 10-15 users and about 200 resources seems to be sufficient for a precision of about 0.75 when compared to the results from resource co-ocurrence. This result was determined by implementing custom tools which could automatically query our web services, gather and average the results for various numbers of users and of resources. A more extensive description can be found in [13].

The table in fig. 2 shows that none of the similarity measures was able to reliably discover synonyms. We concluded that this was due to the fact that our test set didn't contain terms which have widely used synonyms. Another set of 5 terms was created that did have clear synonyms (e.g. CPU/processor). These additional terms were then used to re-evaluate the cosine-based measures to see whether they would reliably return synonyms, because the literature strongly suggests that they are suited for this task. In some cases, these measures can indeed be used to identify synonyms. However, their position in the result list is unreliable. For example, if we consider the differences in lowercase/uppercase, the same tags appear in different forms in the repository (e.g. 'java', 'Java' etc.). We would expect that given a tag 'java', the other form 'Java' also appears

(high) in the list. In exactly 50% of the cases, we find a tag in the related tags list (somewhere in the top-20), which only differs in uppercase/lowercase. The position in the top-20 ranges from 3rd to 20th.

In order to improve the ranking of familiar or unfamiliar concepts, we also did experiments which take the term frequency and inverse document frequency into account (i.e. tf versus tf * idf). After an evaluation based on our 12 test terms, we concluded that the results of both methods were almost similar. We did not find any advantage using tf * idf.

5 Reference Ontologies

Even though the application of the various similarity measures didn't allow for an automatic interpretation of the data in the computing domain, we have used it as first step in the ontology enrichment process. More specifically, given a seed tag in the LT4eL computing ontologies, similarity measures can be employed to find additional related tags that can be used to enhance the ontology.

The main goal is to include information that is relevant to a learner and his peers in the existing domain ontology structure. Tagging systems provide us with a domain vocabulary which is validated as common knowledge by the community that has produced it. The information implicitly contained in tag collections can be employed to assess how relevant a term is in a given domain. Similarity measures allow us to select possible lexicalizations of concepts which are related to the existing ones in the ontology, and which we consider to be 'socially relevant' with respect to the input lexicalisation. More specifically, we have chosen to adopt the resource coocurrence measure in our system for efficiency reasons and wide use in the literature.

However, if we want to map the related terms identified by similarity measures to the concepts present in the ontology, we still face the problem of identifying the appropriate relations. To this end, several heuristics are employed. They heavily rely on the use of a large knowledge base such as DBpedia.

For example, we employ DBpedia to assess whether a related tag can be considered a new concept or a lexicalization of an existing one. By making use of the SKOS vocabulary [12], we can differentiate between a preferred lexicalization (the head term) and additional lexicalizations (i.e. popular and alternative terms for the same concept). The *rdf:type* assertion between a DBpedia resource and a resource from some other ontology can be used to infer that the DBpedia concept is actually a sub-concept of the object of that statement and should be added as such to the seed ontology.

DBpedia also contains a category structure and a list of all the DBpedia concepts and other categories present in such a category hierarchy. We can automatically calculate the closest shared categories for two concepts and return them.

To summarize with an example: given the pre-existing domain ontology concept 'XHTML', the similarity measure system generates the tag 'xslt' which is attested in DBpedia as a resource (i.e. a concept) and it shares the category

'XML' with the 'XHTML' concept. Given that the category 'XML' is already a concept present in the domain ontology the new concept 'XSLT' can be added as a subclass of it.

The resulting ontology integrates the socially relevant concepts within the structure of an expert view domain ontology. Methods that derive ontology-like structures from tag systems such as those described in section 2 cannot provide high quality of results. This is due to the unavailability of explicit structural information in folksonomies. On the contrary, this structural information has been made explicit in ontologies and our approach relies on it.

6 Tag Disambiguation

The ontology enrichment approach discussed in the previous sections can be employed to enrich an ontology with unambiguous terms such as 'HTML' while this is not the case for 'Java' (both a programming language and an island in Indonesia amongst other things).

The ontology enrichment pipeline is employed to enhance an ontology in the computing domain which is relatively unambiguous. However, even in this domain several exceptions are attested. For example, out that of the 7231 tags, resulting as output of similarity measures, only 1271 are unambiguous while 5960 are ambiguous. In the latter case, disambiguation is crucial in order to properly map tags to concept.

An interesting approach to disambiguation is proposed in [16]. They use Tagpedia, which is a system based on a Wikipedia corpus in order to disambiguate tags. More specifically, disambiguation is carried out by relying on the distance in the text between two tags. An alternative approach is described in [5], in which the agreement between two concepts is not calculated directly, due to efficiency reasons. He considers instead the common Wikipedia categories related to the ambiguous terms as a way to disambiguate.

In our approach, we make use of the Wikipedia disambiguation page to obtain possible interpretations of a term. We rely also on the structural information (i.e. pagelinks) that is available in Wikipedia in order to construct a network of interrelated meanings. In Wikipedia, pagelinks specify relations among various articles that, for our purpose, we consider concepts. The basic assumption behind our disambiguation approach is that there is a correspondence between the various tags that people associate with resources in social media applications and the pagelink structure in Wikipedia.

We have thus deviated from the work in [5] which is dependent on Wikipedia category information for disambiguation. This implies that we are in a position to disambiguate concepts which lack proper categorisation. We believe that this should be more suitable for our task because it can deal with incompatible sets of meanings.

To exemplify our approach to disambiguation, consider a user that has tagged a resource with the tags: *python* and *ruby*. Both *ruby* and *python* are ambiguous terms, where *ruby* could refer to things such as an expensive jewel or a

programming language and *python* could mean a specific species of snake or a programming language. By considering the different concepts (i.e. Wikipedia articles) associated with the terms *python* and *ruby* and pagelinks contained in the articles a choice can be made between the various interpretations. As in previous work, we employ at least two terms in order to disambiguate.

The disambiguation algorithm is illustrated in Figure 3. More specifically, we have developed a new approximation algorithm inspired by social network analysis which is able to disambiguate the tags that are shared by a single tagging instance. We start by retrieving the set of meanings which are associated with each input term (Figure 3, step 1) through the disambiguation page in Wikipedia. A term is considered ambiguous if we retrieve more than one meaning from Wikipedia.

For example, all the possible interpretations for the input terms *C*, *D*, *Ruby*, *Python* and *Perl* are retrieved. The term *Ruby* has interpretations such as: programming language, gem, Ruby MRI, Ruby Wax and hardware design language. *Python* has other interpretations such as: programming language and snake. *C* and *D* are also highly ambiguous. Only the term *Perl* is unambiguous and its only interpretation is a programming language.

After determining the possible concepts for the input term, the pagelinks between these concepts are retrieved.

This graph of interconnected meanings is then processed by a graph layout algorithm (Fruchterman-Reingold) which clusters concepts with strong ties together (Figure 3, step 2).

After the graph layout algorithm has assigned each concept a fixed location in the graph, a Self Organising Map [9] based clustering component is applied. It divides the graph into separate clusters of interconnected concepts (Figure 3, step 3). This step is necessary: considering only unconnected clusters of concepts is not sufficient (as illustrated in the graphical example).

In our example, we obtain a cluster which includes ruby programming language, python programming language and ruby hardware design language, Ruby MRI and Perl. They are in the same cluster because pagelinks occur relatively often between them. On the other hand, ruby gem and python snake will appear far apart from the other concepts, because they lack pagelinks to those concepts.

The centrality of each concept inside its own cluster is calculated next (Figure 3, step 3, nodes with underlined text). Centrality is a measure that indicates which node (concept) is most central or 'important' in its cluster [2]. The centrality values are used to reduce the number of concepts in the cluster. Concepts with the highest centrality value for the associated term are retained. All the others are removed from the cluster.

This means that the concepts Ruby hardware design language and ruby MRI, both originated from the term Ruby, will be removed from the cluster. Only Perl, ruby programming language and python programming language will remain in the cluster after filtering by centrality.

Subsequently the clusters are sorted by the number of concepts still present in them. The idea is that each cluster now contains the maximum amount of

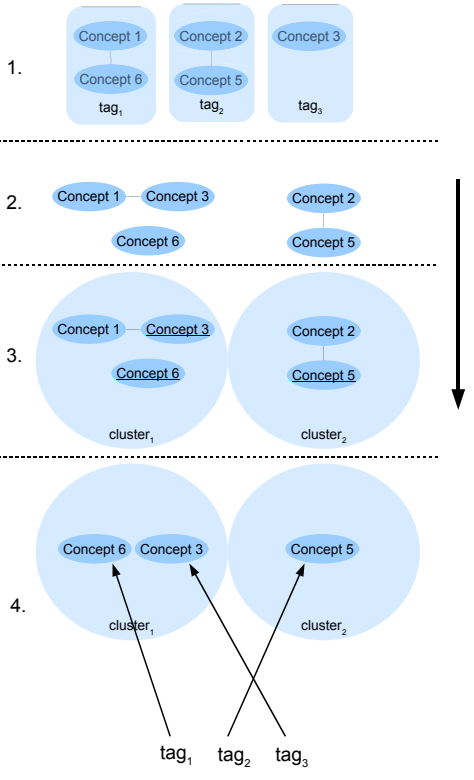

Fig. 3. Disambiguation process

central and coherent meanings for the largest number of terms. Term concept assignments (Figure 3, step 4) will start with the cluster which contains the largest number of concepts (ruby programming language, python programming language, Perl). The concepts from this cluster are paired with their respective input terms and the terms are thus disambiguated.

The next cluster will be chosen by the the number of terms not yet disambiguated. The rationale for not selecting the next-largest cluster is that it could contain conflicting concepts. Disregarding concepts belonging to already disambiguated terms will retain truly complementary clusters instead of conflicting ones. In the example considered, this criterion leaves C and D to be available for disambiguation.

In order to link tags extracted from social media application to their related concepts the following process is applied: For every resource, the tags added by each individual user are considered. Such a collection of tags is called a *Tagging*-instance in the SCOT vocabulary. The tags associated with the Tagging-instance are processed by the disambiguation algorithm and result in a list of term-concept pairs. These term-concept pairs are then stored using the MOAT ontology [15]. It is thus possible to differentiate between *global meanings* as the

list of all meanings that could be related to a tag in a folksonomy space and *personal meanings* related to a specific user or Community of Practice (CoP). As a result not only the meaning of a given tag is available, but also who assigned this meaning.

```
<tag:RestrictedTagging>
  <tag:taggedResource
     rdf:resource="http://www.python.org"/>
    %use a different page in which python is not present
  <foaf:maker
     rdf:resource="http://userdirectory.example.com/Mary"/>
  <tag:associatedTag
     rdf:resource="http://delicious.com/tag/python"/>
  <moat:tagMeaning
     rdf:resource="http://dbpedia.org/resource/Python_(progr_language)"/>
</tag:RestrictedTagging>
```

Listing 1. MOAT example

This possibility is especially relevant in our eLearning application since we can identify the meaning that is common to a group of users sharing a specific interest (i.e. a Community of Practice). The MOAT ontology allows us to model the differences in meaning that emerge in the disambiguation process. The applications in our eLearning domain are obvious. The information can be employed to provide more appropriate search results for the learning material and in addition it becomes possible to identify communities with a superficially similar vocabulary which are actually distinct.

7 Evaluation

In order to evaluate our ontology enrichment methodology, we have compared three different ontologies:

1. the LT4eL computing ontology with the related English lexicon (1200 classes);
2. a manually enriched ontology which takes the LT4eL one as basis (1336 classes and 1672 lexical entries). This is our gold standard.
3. the automatically enriched ontology, which takes the original LT4eL ontology as basis. (2016 classes and 2325 lexical entries)

A first analysis of the lexical differences between (1) and (2) shows a difference of 80 lexicalisations. The aim of our evaluation was to assess whether the automatic enrichment process would add lexicalisations (and related concepts) that overlap with manually added lexicalizations given a similar sub-domain.

The automatically enriched ontology has been generated by considering each coocurring tag in our Delicious data set as eligible for enrichment. Even though we considered every coocurring tag as eligible for use in ontology enrichment, the lexical overlap between the manually enriched ontology and the automatic one is minimal. More specifically, 69 terms which have been added manually

to the LT4eL ontology are multi-word units and are not attested in Delicious. They are representative of the expert view of the domain given their level of specificity and include terms such as: NMTOKEN attribute, XML element type declaration, XML attribute list declaration. The remaining 21 terms are attested in Delicious but only 13 of them are generated by the similarity measures and are attested in DBPedia.

Regardless of the minimal lexical overlap between the manually and the automatic enriched ontology, it is not the case that the terms added automatically are not appropriate and are misplaced in the ontology, as the following evaluation (that filters upper ontology concepts) reveals:

- Total number of unique statements: 1265
- Accurate enough for ontology enrichment: 1010
- Too inaccurate: 255

This brings the amount of usable additions to about 80%. We have analyzed the added relations further and discovered that:

- Relations with the very general *ltfll:related* relation: (598). The 'related' label only indicates that two terms are related but it doesn't say in which way.
  - Correct: 497 (83%)
  - Incorrect: 101
- Clear ontological relations (*rdfs:subclassof* or either DBpedia specific ones): 667
  - Correct:513 (77%)
  - Incorrect: 154

There is minimal overlap between the ontology produced by means of a manual enrichment process carried out by an expert and our automatic enrichment process. The latter includes the vocabulary of the community of users, while the former includes very specialized tags provided by an expert. It is exactly this complementarity that we wanted to achieve by embedding tags into an existing ontology and that we want to exploit in eLearning applications.

In addition to this quantitative evaluation, we have run an experiment focused on support provided by an ontology enhanced with tags in comparison with tag clouds extracted from Delicious (lacking ontological structure) in the context of a learning task. The underlying assumption is that conceptualization can guide learners in finding the relevant information to carry out a learning task (a quiz in our case). The hypothesis was that learners might differ with respect to the way they look for information depending on whether they are beginners or more advanced learners. While beginners might profit from the informal way in which knowledge is expressed through tagging, more advanced learners might profit from the way knowledge is structured in an ontology. In general, both advanced learners and beginners profited from the clear ontology structure present in the graph. However, beginners relied mainly on documents to find the relevant information both in the case of the enhanced ontology and in the case of the cluster of related tags. The advanced learners made more use of the enriched ontology and used less documents. We refer to [13] for additional details on the results of the experiment and their interpretation.

8 Conclusions

One of the goals of the *Language technology for LifeLong learning* project is to develop services that facilitate learners and tutors in accessing formal and informal knowledge sources in the context of a learning task. To this end, a Common Semantic Framework has been developed in which domain ontologies constitute the core element. There are obvious shortcomings in the use of ontologies in this context: they are too static since they model the knowledge of the domain at a given point in time, they might be incomplete or might not correspond to the representation of the domain knowledge available to the learner.

We have proposed an ontology enrichment pipeline to overcome some of these problems by exploiting social data which are crawled from existing social media applications. In our approach, we include the expert view on the domain by maintaining the ontology structure but we complement it with the 'wisdom of the crowd' in order to provide more dynamic ontologies that take into account the evolving vocabulary of the community.

The domain ontology enriched with social tags constitute the basis of the semantic search implemented to retrieve formal and informal resources. However, we have also exploited the possibility of a socially driven search by employing tags and social networks in the recommendation of learning material. Our ultimate goal is to integrate the strong features of both types of searches, that is the structured information contained in ontologies with the social information coming from the tags and social networks. In addition, the creation of ontologies related to a Community of Practice allows for the recommendation of experts that can support learners in their learning tasks.

References

1. Auer, S., Bizer, C., Lehmann, J., Kobilarov, G., Cyganiak, R., Ives, Z.: DBpedia: A nucleus for a web of open data. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007)
2. Brandes, U.: A Faster Algorithm for Betweenness Centrality. Journal of Mathematical Sociology 25(2), 163–177 (2001)
3. Brooks, C.H., Montanez, N.: Improved annotation of the blogosphere via autotagging and hierarchical clustering. In: WWW 2006: Proceedings of the 15th international conference on World Wide Web, pp. 625–632. ACM, New York (2006)
4. Cattuto, C., Benz, D., Hotho, A., Stumme, G.: Semantic grounding of tag relatedness in social bookmarking systems. In: Sheth, A.P., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., Thirunarayan, K. (eds.) ISWC 2008. LNCS, vol. 5318, pp. 615–631. Springer, Heidelberg (2008)
5. Cucerzan, S.: Large-scale named entity disambiguation based on Wikipedia data. In: Proceedings of EMNLP-CoNLL, pp. 708–716 (2007)
6. van Damme, M., Hepp, K., Siorpaes, K.: Folksontology: An integrated approach for turning folksonomies into ontologies. In: Proceedings of the ESWC Workshop Bridging the Gap between Semantic Web and Web 2.0, Innsbruck, Austria. Springer, Heidelberg (2007)

7. Gemmell, J., Shepitsen, A., Mobasher, B., Burke, R.: Personalization in Folksonomies Based on Tag Clustering. In: Intelligent Techniques for Web Personalization & Recommender Systems (2008)
8. Heymann, P., Garcia-Molina, H.: Collaborative Creation of Communal Hierarchical Taxonomies in Social Tagging Systems. Stanford InfoLab Technical Report 2006-10 (2006)
9. Kohonen, T.: The self-organizing map. Neurocomputing 21(1-3), 1–6 (1998)
10. Lemnitzer, L., Simov, K., Osenova, P., Mossel, E., Monachesi, P.: Using a domain-ontology and semantic search in an eLearning environment. In: Proceedings of The Third International Joint Conferences on Computer, Information and Systems Sciences, and Engineering (CISSE 2007). Springer, Heidelberg (2007)
11. Mika, P.: Ontologies are us: A unified model of social networks and semantics. Journal of Web Semantics 5(1), 5–15 (2007)
12. Miles, A., Matthews, B., Wilson, M., Brickley, D.: SKOS Core: Simple knowledge organisation for the web. In: Proceedings of the International Conference on Dublin Core and Metadata Applications, pp. 12–15 (2005)
13. Monachesi, P., Markus, T., Mossel, E.: Ontology Enrichment with Social Tags for eLearning. In: Cress, U., Dimitrova, V., Specht, M. (eds.) EC-TEL 2009. LNCS, vol. 5794, pp. 385–390. Springer, Heidelberg (2009)
14. Monachesi, P., Simov, K., Mossel, E., Osenova, P.: What ontologies can do for eLearning. In: Proceedings of International Conference on Interactive Mobile and Computer Aided Learning, IMCL 2008 (2008)
15. Passant, A., Laublet, P.: Meaning of A Tag: A collaborative approach to bridge the gap between tagging and Linked Data. In: Proceedings of the WWW 2008 Workshop Linked Data on the Web (LDOW 2008), Beijing, China (2008)
16. Ronzano, F., Marchetti, A., Tesconi, M., Minutoli, S.: Tagpedia: a semantic reference to describe and search for web resources. In: Proc. of The Workshop Social Web and Knowledge Management of the World Wide Web Conference, vol. 8, pp. 19–25 (2008)
17. Schmitz, P.: Inducing Ontology from Flickr Tags. In: Proceedings of the Collaborative Web Tagging Workshop at the 15th WWW Conference (WWW 2006), Edinburgh, Scotland (2006)
18. SigurbjŽrnsson, B., van Zwol, R.: Flickr tag recommendation based on collective knowledge. In: Proc. 17th Intl. Conf. on World Wide Web, pp. 327–336 (2008)
19. Specia, L., Motta, E.: Integrating Folksonomies with the Semantic Web. In: Franconi, E., Kifer, M., May, W. (eds.) ESWC 2007. LNCS, vol. 4519, pp. 624–639. Springer, Heidelberg (2007)
20. Wu, X., Zhang, L., Yu, Y.: Exploring Social Annotations for the Semantic Web. In: Proc. of WWW 2006 (2006)