Department of
Information Engineering
and Computer Science DISI

UNIVERSITY
OF TRENTO - Italy

DISI - Via Sommarive 14 - 38123 Povo - Trento (Italy)
http://www.disi.unitn.it

SENSE INDUCTION IN FOLKSONOMIES

Pierre Andrews and Juan Pane and Ilya Zaihrayeu

April 2011

Technical Report # DISI-11-459

Sense Induction in Folksonomies *

Pierre Andrews and **Juan Pane** and **Ilya Zaihrayeu**
DISI - University of Trento, Italy
{andrews,pane,ilya}@disi.unitn.it

Abstract

Folksonomies, often known as tagging systems, such as the ones used on the popular Delicious or flickr websites, use a very simple knowledge organisation system. Users are thus quick to adopt this system and create extensive knowledge annotations on the Web. However, because of the simplicity of the folksonomy model, the semantics of the tags used is not explicit and can only be inferred from the context of use of the tags. This is a barrier for the automatic use of such knowledge organisation systems by computers and new techniques have to be developed to extract the semantic of the tags used. In this paper we discuss an algorithm to detect new senses of terms in a folksonomy; we also propose a formal evaluation methodology that will enable to compare results between different approaches in the field.

1 Introduction

Folksonomies are uncontrolled knowledge organisation systems where users can use free-text tags to annotate resources. They create a network of user-tag-resource triplets that encodes the knowledge of users [Mika, 2007]. However, because they are based on the use of free-text tags, folksonomies are prone to language ambiguity issues as there is no formalisation of polysemy/homography (where one tag can have multiple senses) and synonymy (where multiple tags can have the same sense) [Golder and Huberman, 2006]. This lack of explicit semantics makes it difficult for computer algorithms to leverage the whole knowledge provided by the folksonomy model.

While there are existing Word Sense Disambiguation (WSD) algorithms in the state of the art, they are not completely adapted to folksonomies. WSD algorithms use an existing vocabulary to link terms (in our case tags) to concepts, thus discovering the semantics of the tags used. However, as shown in [Andrews *et al.*, 2011], a standard a structured vocabulary such as WordNet [Miller, 1998] covers less that 50% of the terms used by the users of the folksonomy. This happens because of the dynamic nature of folksonomies where new concepts and terms appear quickly. To tackle this issue, *sense induction* algorithm are being developed [García-Silva *et al.*, 2010] to detect new concepts and extend the existing vocabularies.

While the computer does not "know" the actual meaning of the free-text tag used, the users always know the meaning they wanted to use when they tagged a resource. So if they tagged a bookmark with "java", in their mind, at the time of tagging, they knew exactly if they meant the "indonesian island" or the "programming language". This principle has already been widely illustrated in the automatic ontology building field where social network analysis methods were introduced [García-Silva *et al.*, 2010] to extract the so-called "emergent semantics" [Aberer *et al.*, 2004].

In this article, we discuss our approach to the detection of new concepts in folksonomies and how it differs from the state of the art by enabling the detection of homographs (see Sections 2 and 3). Because the state of the art also lacks a formalised evaluation methodology, we discuss in Section 4 a possible approach for a comparable and reproducible evaluation. While we are currently applying this evaluation methodology to the algorithm we introduce, we are not reporting results in this paper as they are not yet available.

2 Sense Induction

The method used to extract the semantics from folksonomies is what is called *tag clustering* and its principle is based on machine learning clustering algorithms [Xu and Wunsch, 2005]. This clustering is based on the principle that similar tags will have the same meaning and can thus be attached to the same *"concept"* in the created vocabulary. For instance, if the algorithm finds out that "opposition" and "resistance" are similar, then it can associate it to one concept for that meaning. One of the main issues is thus to compute the similarity between tags to run the clustering algorithms that will attach similar tags together. To do this, all the methods available currently use a mix of measures based on the collocation of tags on resources and their use by users. If two tags are often used by the same user on different resources or by different users on the same resource, then they can be considered similar [García-Silva *et al.*, 2010].

*This work has been partially supported by INSEMTIVES project (FP7-231181, see http://www.insemtives.eu).

This assumption on the computation of the similarity of tags is, in our opinion, one of the first weak points of these approaches as it makes the assumption that one tag can only have one meaning. Thus these algorithms can find synonyms of the most popular sense but cannot deal with the polysemy of the words. For example, if the tag "java" is collocated with "indonesian island" on 200 resources and with "programming language" on 1000 resources, then it will be considered to be similar to the latter and the fact that it has a second meaning is lost. However, [Zhang *et al.*, 2006] show that tags are often ambiguous in folksonomies (their study is also based on Delicious[1]) and can bare more than one meaning. In the algorithm we propose, we add an extra step to the clustering to first identify the diverse senses of polysemous tags and in the following clustering steps, we do not consider tags directly, but the unique senses that they can take (see Section 3).

3 Algorithms

We propose to adopt a parametric based clustering approach slightly different from the standard KMeans and KNN algorithms that are often discussed in the state of the art of ontology construction from folksonomy (see, for a review [García-Silva *et al.*, 2010]). In fact, these algorithms, while being the most popular in the clustering field, are not well tailored to our application domain as they take as an input-parameter the number of expected clusters (the K in the name). The state of the art approaches on ontology building from folksonomies cluster all the tags together to find all the concepts that they represent (see figures two and four in the review of Garcia-Silva et al. [García-Silva *et al.*, 2010]). In this case, they can optimise the K parameter to find the best overall number of clusters for their dataset. However, in our approach, we have added an extra step where clustering is applied to detect the different senses in which one tag can be used. In this case, we cannot find an overall optimal value for the number of clusters to look for as each term might have a different number of senses.

Thus, we need to use a clustering algorithm that can work without this parameter as input. We use the DBScan algorithm [Ester *et al.*, 1996] to do a density based clustering. This approach to clustering has various advantages for our application:

- it does not require as input the number of clusters to be found. Instead it takes two parameters: ϵ, the minimum distance between two items to put them in the same cluster and m the minimum number of items in a cluster.

  ϵ is easier to optimize in our use case than to compute the K parameter as we can find it by studying the accuracy of each clustering step as discussed in Section 4.

- while the KMean and KNN algorithms assign all items in the clustering space to a cluster, the DBScan algorithm can decide that some of the items to be clustered are noise and should not be considered. This is very important in our application domain as it allows for leaving out very personal or subjective uses of a term that might not be aligned with the rest of the community understanding of the term; and
- the DBScan algorithm can detect clusters that have more complex "shapes" than the standard hyperspherical clusters returned by vector quantization based clustering such as the KMeans and KNN [Xu and Wunsch, 2005].

[1] `http://www.delicious.com`

While there is already some research done on diverse similarity measures applicable to concept detection and learning in the Natural Language Processing field (for instance [Alfonseca and Manandhar, 2002a] or [Jamoussi, 2009]), the existing clustering techniques discussed in the folksonomy field are only considering raw collocation counts (of tags, resources or users) as a similarity measure between tags. For instance, [Alfonseca and Manandhar, 2002a] proposes to combine four different measures to compute sense similarities: the *topic signature*, the *subject signature*, the *object signature* and the *modifier signature*. While most of these measures can only be applied to textual documents as they require to know noun-verb relationships in a sentence, the *topic signature* is interesting in the domain of folksonomy where one of the only context we have for computing the distances is the list of collocations. However, these collocations can be considered and weighted in different ways and [Jamoussi, 2009] points out that simple vector distances or cosinus distances between *topic signatures* are not always powerful enough. The authors show that information based measures – such as the Kullback-Leibler divergence of word distribution, the mutual information – can be used to have more powerful measures of semantic distances between concepts based on the Distributional Semantics principles [Lin, 1998]. The authors of [Weinberger *et al.*, 2008] have proven that this measure can be applied with success to the domain of folksonomies to disambiguate tag senses.

For the algorithm that we discuss in this section we use clustering algorithms relying on distance measures between User-Resource pair and between tag senses. We are currently experimenting with different measures, from the standard tag collocation measures proposed in the current state of the art to the more advanced distributional measures described above.

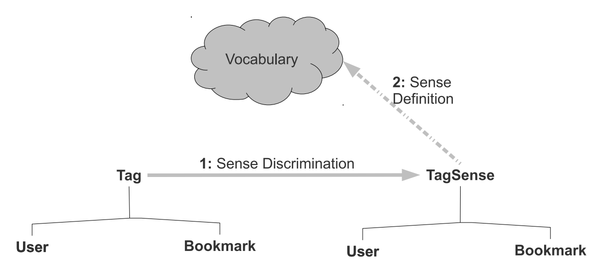

Figure 1: Sense-User-Bookmark Tripartite graph

To enrich the structured vocabulary with a new concept from a free-text tag, we propose to do the concept detection in three stages:

1. For each tag, we cluster the user-resource bipartite graph that are attached to this tag. By doing so, as was hinted by [Au *et al.*, 2007], we discover the different meanings of the tag. By considering each cluster to be a tag

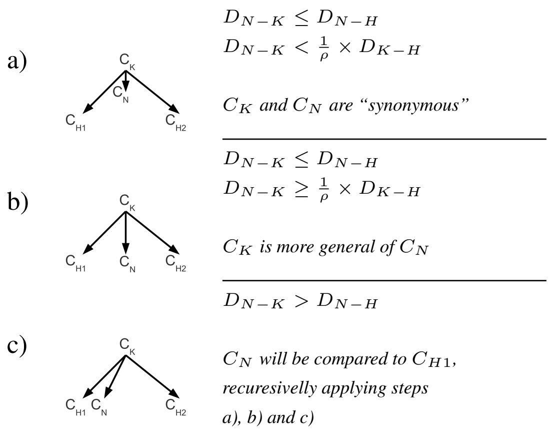

Figure 2: Decisions to Extend the Concept Taxonomy

sense, we replace the tag in the user-resource-tag tripartite graph by its senses and the tripartite graph becomes a user-resource-sense graph as illustrated in Figure 1. In this way, if we consider our previous example, the tag "java" will be split in two senses: `java-1`, similar to "indonesian island" and `java-2`, similar to "programming language".

2. We then apply the same principle as the one discussed in the state of the art on the user-resource-sense tripartite graph to cluster similar senses together (see [García-Silva *et al.*, 2010] for a review).

3. Once the tag senses have been clustered together, we identify new concepts for each of the clusters. This process is equivalent to finding the relation (in particular hypernym/hyponym relations) of the new concept (represented by the cluster of tag senses) in the structured vocabulary. This can be achieved by applying a hierarchical classification approach similar to the one proposed in [Alfonseca and Manandhar, 2002a]. In their approach to ontology building, they consider a similarity measure between a *known* concept C_k in the vocabulary and a new concept C_n. If the distance between these two concepts is smaller than the distance between C_n and any of the hyponyms of C_k, then C_n is considered to be the hyponym of C_k. Otherwise, they continue the search down the conceptual hierarchy. We alter this approach by splitting it in three cases as we believe that there can also be cases in which the new concepts C_n are actually synonyms of an existing concept C_k. The updated solution is as follows:

   - if the new concept C_n is closer to the existing concept C_k than to any of the its hyponyms, but much more – this is defined by the parameter ρ as defined in Figure 2a) – similar to C_k than any of its hyponyms, then it is most likely that C_n is a synonym of C_k (Figure 2a)[2]);
   - if the new concept C_n is closer to the existing concept C_k than to any of the its hyponyms, but not much more similar to C_k than any of its hyponyms, then it is most likely that C_n is more specific than C_k (Figure 2b));
   - if the new concept C_n is closer to the a hyponyms of C_k than C_k, then we recursively apply these three steps to this most similar hyponym (Figure 2c));

   We apply this search procedure on our structured vocabulary (in our case WordNet), starting from the root of its conceptual *is-a* hierarchy.

   This approach is parametric as it depends on the value of ρ, which specifies the threshold to decide if a new concept is more specific than an existing concept or is just a synonymous. This parameter will be different depending on the specific application domain and will decide how much specific the structured vocabulary will get.

[2]where D_{i-j} is the distance between C_i and C_j.

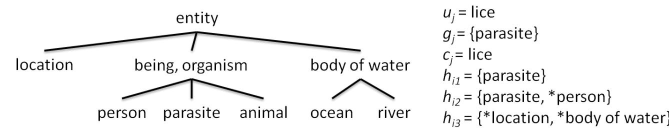

Figure 3: Example of taxonomy, an unknown relevant concept u_j, its correct generalisations g_j and the generalisations proposed by three hypothetical algorithms h_{ik}

We are currently running evaluations to show the behaviour of these algorithms with different values of the ϵ, m and ρ parameters and will report on these in future publications.

4 Evaluation Methodology

While there is existing research on the automatic construction of ontologies from folksonomies, [García-Silva *et al.*, 2010] points out that there is not yet any agreed evaluation dataset. In fact, from our knowledge of the state of the art approaches, there is not yet an appropriate evaluation methodology in the field. This is mostly due to the lack of a gold standard evaluation dataset and thus the evaluation of the existing methods were often only evaluated "*subjectively*" [Lin *et al.*, 2009] by checking manually some extracted clusters, thus only providing anachronyc results that cannot be compared or reproduced [García-Silva *et al.*, 2009; Van Damme *et al.*, 2007; Specia and Motta, 2007].

However, as pointed out earlier, the NLP field has already tackled the issue of concepts extraction from text and has considered different evaluation measures for this task. [Alfonseca and Manandhar, 2002b] describes the evaluation problem as follows:

> Let us suppose that we have a set of unknown concepts that appear in the test set and are relevant for a specific domain: $U = \{u_1, u_2, \ldots, u_n\}$. A human annotator has specified, for each unknown concept u_j , its maximally specific generalisations from the ontology: $G_j = \{g_{j,1}, g_{j,2}, \ldots, g_{j,m_j}\}$.
>
> Let us suppose that an algorithm decided that the unknown concepts that are relevant are $C = \{c_1, c_2, \ldots, c_l\}$. For each C_i, the algorithm has to

provide a list of maximally specific generalisations from the ontology: $H_i = \{h_{i,1}, h_{i,2}, \ldots, h_{i,p_i}\}$. (See Figure 3, adapted from [Alfonseca and Manandhar, 2002b])

From this definition, a number of evaluation metrics can be computed:

Accuracy the amount of correctly identified maximally specific generalisations,

Parsinomy the amount of concepts for which a correct set of generalisations is identified,

Recall the amount of concepts that were correctly detected and to which at least one relevant maximally specific generalisation was found,

Precision the ratio of concepts that were correctly attached to their maximally specific generalisations to the total of concepts identified

Production the amount of proposed maximally specific generalisations per concept.

Learning Accuracy the distance, in the concept hierarchy, from the concept proposed placement to its true placement (from [Hahn and Schnattinger, 1998]).

As can be seen from these proposed measures, a gold standard needs to be available that provides the "maximally specific generalisations" (G_j) for each concept (U). Alfonseca and Manandhar [Alfonseca and Manandhar, 2002b] use a dataset of textual documents that is manually annotated for this purpose. However, we need to evaluate the algorithm within a folksonomy and thus we use the dataset described in [Andrews *et al.*, 2011] (the tags2con dataset) as it provides a manually validated disambiguation for each tag in a subset of the Delicious folksonomy. The tags2con dataset is a collection of bookmarks from the Delicious website for which each free-text tag associated to the bookmarks has been manually disambiguated to its corresponding concept in a structured vocabulary, in this case WordNet.

The measures listed above can be computed on this dataset by applying a leave one out approach to the evaluation. That is, we iterate through all tag annotations already linked to a concept in the gold standard; we "forget" the senses of one tag at a time and apply the algorithm on this tag; we then compare the detected senses and their new place in the taxonomy for this tag to the actual sense that the gold standard defines.

While this is a possible evaluation procedure to evaluate the final output of the whole algorithm, the current dataset is not in a form that allows for the evaluation of the intermediate results. In particular, to optimise the ϵ and m parameters of the clustering steps, we have to be able to evaluate independently the accuracy of each stage of the algorithm. In the same way, we need to be able to evaluate the distance metrics used and compare different approaches. For this, we need a clustering gold standard, that provides the "true cluster" (class) of each user-resource pairs in the dataset so that we compare the found clusters to this gold standard results. In the following paragraphs we discuss a strategy to generate such a clustering gold standard.

When building the gold standard (GS^j) we want to automatically generate the set of unknown concepts (U and C_i) to

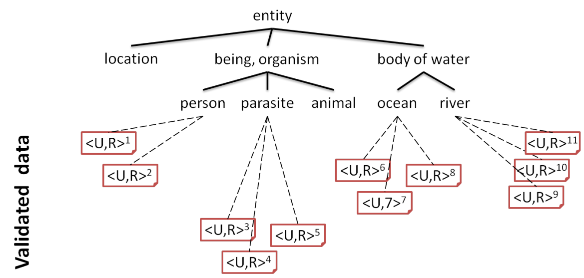

Figure 4: Validated Data

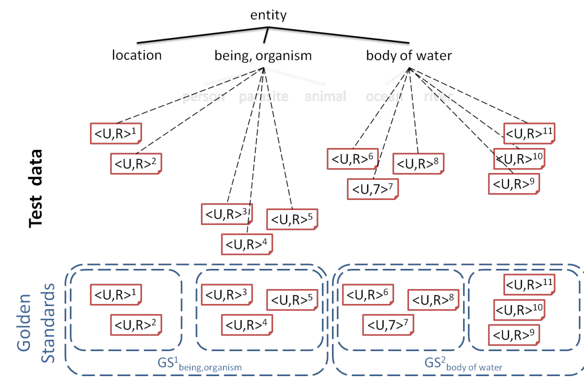

Figure 5: One Possible Test Set

be clustered, their classes, and the maximally specified generalization G_j. In order to do so, we perform the following steps:

1. we define G_j to be a concept in our Structured Vocabulary (SV) for which there is more than one hyponym that has more than one manually validated associated term in the annotation. In the example in Figure 4, the concept G_1 ="being, organism" has two hyponyms ("person" and "parasite") that contain more than one annotation attached to them, also the concept G_2 ="body of water" has two two hyponyms ("ocean" and "river") that have more than one annotation attached to it. Each of

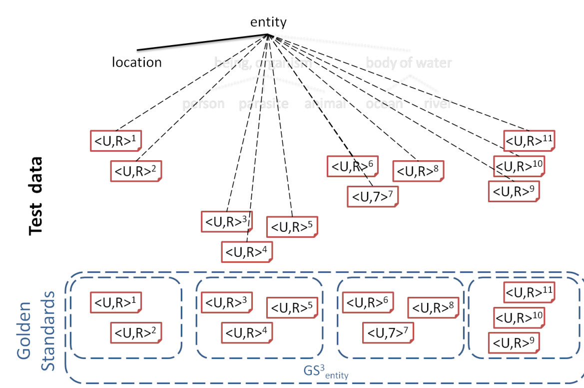

Figure 6: Another Possible Test Set Generated from a Higher Concept in the CV

GS^k	C, U	G_j	Clusters and new concepts
$GS^1_{being,organism}$	$C = U =$ {person, parasite}	$G_1 =$ {"being, organism"}	person = $\langle U, R\rangle^1, \langle U, R\rangle^2$ parasite = $\langle U, R\rangle^3, \langle U, R\rangle^4, \langle U, R\rangle^5$
$GS^2_{bodyofwater}$	$C = U =$ {ocean, river}	$G_2 =$ {"body of water"}	ocean = $\langle U, R\rangle^6, \langle U, R\rangle^7, \langle U, R\rangle^8$ river = $\langle U, R\rangle^9, \langle U, R\rangle^10, \langle U, R\rangle^{11}$
GS^3_{entity}	$C = U =$ {person, parasite, ocean, river}	$G_3 =$ {"entity"}	person = $\langle U, R\rangle^1, \langle U, R\rangle^2$ parasite = $\langle U, R\rangle^3, \langle U, R\rangle^4, \langle U, R\rangle^5$ river = $\langle U, R\rangle^9, \langle U, R\rangle^10, \langle U, R\rangle^{11}$ river = $\langle U, R\rangle^9, \langle U, R\rangle^10, \langle U, R\rangle^{11}$

Table 1: Resulting gold standards GS^k for the evaluation of the sense induction algorithm.

the complying concepts ("being, organism" and "body of water") will generate a set of clusters for the gold standard datasets $GS^1_{being,organism}$ and $GS^2_{bodyofwater}$.

2. we "forget" momentarily that each of the hyponyms of G_j exist. Since for each of these children C_i we know their corresponding annotations, we create a class for each deleted concept, and define the boundary of the GS^k clusters to these particular classes. In our example in Figure 5, we can see that two gold standards have been created: GS^1 for "being, organism" and GS^2 for "body of water", each of them containing two clusters (one for each deleted concept).

3. Starting from the leaves, we recursively repeat the process by further "forgetting" concepts higher in the hierarchy and thus creating more gold standard sets of increasing difficulty as the higher we go in the hierarchy, the more classes will be created in GS^k. In our example in Figure 6, we further "forget" the concepts "being, organism" and "body of water" and create another gold standard GS^3 for "entity", creating four clusters.

If we apply the above mentioned process on the dataset depicted in Figure 4 we obtain three GS datasets as shown in Table 1. When using the tags2con dataset [Andrews *et al.*, 2011], we have 4 427 gold standard annotations representing manually validated user-bookmark-tagsense triplets, from these we build 857 different GS at various depth of the WordNet *is-a* graph. We are currently running the evaluation of different distances on this gold standard.

The purpose of each gold standard GS^k is twofold:

1. Evaluate Step one of the sense induction algorithm presented in the previous Section 3, where the input is a set of free-text tags C and the output is a set of clusters of similar tags that represent a new concept. In our example in Figure 4, we would be calling the clustering algorithm with each of the gold standard GS^k. Then, to compute the accuracy of the clustering, we compare the produced results H_i with the classes of the gold standard with standard cluster evaluation metric such as Purity, Accuracy and Precision/Recall [Amigó *et al.*, 2009].

2. Considering that we know the parent concept G_j for each gold standard GS^k, we also evaluate Step three of the sense induction algorithm where for each cluster produced, a new concept also has to be added to the SV as more specific than an existing concept in the SV. The generalisations in the SV discovered by the algorithm (H_i) is be compared to the one given in the gold standard (G_j). In our example in Figure 4, if we pass GS^1 to the algorithm, it should create concepts for "person" and "parasite", and put them as hyponyms of "being, organism".

5 Results

Using the methodology described in the previous section, we have run a set of preliminary evaluation for the first step of the algorithm using different clustering distances found in the state of the art.

To compute the minimum baseline, we perform random runs where a random number of clusters between one and the number of instances is selected and each instance is assigned randomly to one of these cluster. The mean F-measure on one thousand runs is of 25.8%[3].

In the state of the art, the number of collocated tags between bookmarks, and the number of users using a tag are most often used to compare bookmarks or tags. We have thus started by evaluating these distance measures to establish the state of the art baseline. When using only tag collocation, the first step clustering algorithm can achieve a maximum F-measure of 59.7%[4]. The user collocation measure achieves a very similar result with a maximum F-measure of 59.1%[5], we can however see that the distribution between precision and recall of these two approaches is quite different.

We are currently running evaluations for the other steps of the algorithm and with more complex distance measures that should improve on the naive tag collocation approaches.

6 Conclusion and Future work

We have presented a novel approach to detect concepts in a folksonomy. Our approach is an extension to the state of the art that adds a method to detect polysemous/homograph tags and assign them to different senses. Because of the – acknowledged – lack of standard evaluation methodology in the state of the art, we also propose a new methodology for evaluating sense induction in a folksonomy and building datasets to run such evaluation.

[3] SD = 27.9%; Precision=42.2%, Recall=29.2%.

[4] Precision=59.4%, Recall=63.4%.

[5] Precision=64.8%, Recall=40.1%.

We are currently running evaluations of different distance metrics and parameters to our algorithms by applying the proposed evaluation methodology described here and will report on results of this new approach in upcoming publications[6].

References

[Aberer *et al.*, 2004] Karl Aberer, Philippe C. Mauroux, Aris M. Ouksel, Tiziana Catarci, Mohand S. Hacid, Arantza Illarramendi, Vipul Kashyap, Massimo Mecella, Eduardo Mena, Erich J. Neuhold, and Et. Emergent Semantics Principles and Issues. In *Database Systems for Advances Applications (DASFAA 2004), Proceedings*, Lecture Notes in Computer Science, pages 25–38. Springer, March 2004.

[Alfonseca and Manandhar, 2002a] Enrique Alfonseca and Suresh Manandhar. Extending a lexical ontology by a combination of distributional semantics signatures. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*, EKAW '02, pages 1–7, London, UK, 2002. Springer-Verlag.

[Alfonseca and Manandhar, 2002b] Enrique Alfonseca and Suresh Manandhar. Proposal for evaluating ontology refinement methods. *Language Resources and Evaluation*, 2002.

[Amigó *et al.*, 2009] Enrique Amigó, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, 12:461–486, 2009. 10.1007/s10791-008-9066-8.

[Andrews *et al.*, 2011] Pierre Andrews, Juan Pane, and Ilya Zaihrayeu. Semantic disambiguation in folksonomy: a case study. In *Advanced Language Technologies for Digital Libraries*, Lecture Notes on Computer Science (LNCS) Hot Topic subline. to Springer-Verlag, 2011.

[Au *et al.*, 2007] Au, N. S. Gibbins, and N. Hadbolt. Understanding the Semantics of Ambiguous Tags in Folksonomies. In *The International Workshop on Emergent Semantics and Ontology Evolution (ESOE2007) at ISWC/ASWC 2007*, November 2007.

[Ester *et al.*, 1996] Martin Ester, Hans peter Kriegel, Jrg S, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pages 226–231. AAAI Press, 1996.

[García-Silva *et al.*, 2009] A. García-Silva, M. Szomszor, H. Alani, and O. Corcho. Preliminary results in tag disambiguation using dbpedia. In *Proc. of KCAP*, USA, 2009.

[García-Silva *et al.*, 2010] A. García-Silva, O. Corcho, H. Alani, and A. Gómez-Perez. Review of the state of the art: Discovering and associating semantics to tags in folksonomies. *The Knowledge Engineering Review*, 2010.

[6] while the results are not ready at the time of submitting this paper for review, they will be available at the time of the workshop.

[Golder and Huberman, 2006] Scott Golder and Bernardo A. Huberman. The structure of collaborative tagging systems. *Journal of Information Science*, 32(2):198–208, April 2006.

[Hahn and Schnattinger, 1998] Udo Hahn and Klemens Schnattinger. Towards text knowledge engineering. In *IN AAAI/IAAI*, pages 524–531, 1998.

[Jamoussi, 2009] Salma Jamoussi. Une nouvelle reprsentation vectorielle pour la classification smantique. In *Apprentissage automatique por le TAL*, volume 50, pages 23–57. TAL, 2009.

[Lin *et al.*, 2009] Huairen Lin, Joseph Davis, and Ying Zhou. An integrated approach to extracting ontological structures from folksonomies. In Lora Aroyo, Paolo Traverso, Fabio Ciravegna, Philipp Cimiano, Tom Heath, Eero Hyvnen, Riichiro Mizoguchi, Eyal Oren, Marta Sabou, and Elena Simperl, editors, *The Semantic Web: Research and Applications*, volume 5554 of *Lecture Notes in Computer Science*, pages 654–668. Springer, Berlin / Heidelberg, 2009.

[Lin, 1998] Dekang Lin. Automatic retrieval and clustering of similar words. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 2*, ACL '98, pages 768–774, Stroudsburg, PA, USA, 1998. Association for Computational Linguistics.

[Mika, 2007] Peter Mika. Ontologies are us: A unified model of social networks and semantics. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(1):5–15, March 2007.

[Miller, 1998] G. Miller. *WordNet: An electronic Lexical Database*. MIT Press, 1998.

[Specia and Motta, 2007] L. Specia and E. Motta. Integrating folksonomies with the semantic web. In *Proc. of the European Semantic Web Conference (ESWC2007)*, volume 4519 of *LNCS*, pages 624–639, Berlin Heidelberg, Germany, July 2007. Springer-Verlag.

[Van Damme *et al.*, 2007] Caline Van Damme, Martin Hepp, and Katharina Siorpaes. Folksontology: An integrated approach for turning folksonomies into ontologies. In *Bridging the Gep between Semantic Web and Web 2.0 (SemNet 2007)*, pages 57–70, 2007.

[Weinberger *et al.*, 2008] Kilian Quirin Weinberger, Malcolm Slaney, and Roelof Van Zwol. Resolving tag ambiguity. In *Proceeding of the 16th ACM international conference on Multimedia*, MM '08, pages 111–120, New York, NY, USA, 2008. ACM.

[Xu and Wunsch, 2005] Rui Xu and Donald Wunsch. Survey of clustering algorithms. *IEEE Transactions on Neural Networks / a Publication of the IEEE Neural Networks Council*, 16(3):645–678, May 2005. PMID: 15940994.

[Zhang *et al.*, 2006] Lei Zhang, Xian Wu, and Yong Yu. Emergent Semantics from Folksonomies: A Quantitative Study. In *Journal on Data Semantics VI*, Lecture Notes in Computer Science, chapter 8, pages 168–186. Springer, 2006.