Folksonomized Ontologies – from social to formal

Hugo Alves
Instituto de Computação - Unicamp
Avenida Albert Einstein, 1251
Cidade Universitária, Campinas, Brazil
hugo.alves@students.ic.unicamp.br

André Santanchè
Instituto de Computação - Unicamp
Avenida Albert Einstein, 1251
Cidade Universitária, Campinas, Brazil
santanche@ic.unicamp.br

ABSTRACT

An ever-increasing number of web-based repositories aimed at sharing content, links or metadata rely on tags informed by users to describe, classify and organize their data. The term folksonomy has been used to define this "social taxonomy", which emerges from tagging carried by users interacting in social environments. It contrasts with the formalism and systematic creation process applied to ontologies. In our research we propose that ontologies and folksonomies have complementary roles. The knowledge systematically organized and formalized in ontologies can be enriched and contextualized by the implicit knowledge which emerges from folksonomies. This paper presents our approach to build a "folksonomized" ontology as a confluence of a formal ontology enriched with social knowledge extracted from folksonomies. The formal embodiment of folksonomies has been explored to empower content search and classification. On the other hand, ontologies are supplied with contextual data, which can improve relationship weighting and inference operations. The paper shows a tool we have implemented to produce and use folksonomized ontologies. It was used to attest that searching operations can be improved by this combination of ontologies with folksonomies.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

Keywords

folksonomy, ontology, semantic similarity, information content

1. INTRODUCTION

The popularization of web-based systems offering services for content storage, indexation and sharing fostered a rapid growth of content available on-line. There are more than 5 billion images hosted on Flickr[1] and more than 180 million URL addresses on Delicious[2]. These systems increasingly rely on tag-based metadata to organize and index all the amount of data. The tags are provided by users usually connected in social networks, who are free to use any word as tag; there is no central control. The term folksonomy – combining the words "folk" and "taxonomy" [21] – has been used to characterize the product which emerges from this tagging in a social environment.

Any operation involving indexation, classification or discovery of content in these web-based systems will require a comparison among the involved tags. In this topic, there are approaches ranging from a pure lexical or statistical comparison of words to a richer semantic analysis of relations, by associating tags to formal ontologies. In many contexts, this semantic directed approach will enable machines to better classify, rank, disambiguate and discover tags, enriching the systems and the user experience. Recent investigations explore this relationship in different directions, for example: (i) by deriving ontologies from folksonomies [18, 20]; (ii) by manually or automatically connecting tags to ontologies [1, 7]. In either case, there is still a unidirectional perspective, in which a model takes advantage of the other.

This work addresses a fusion perspective. The proposed *folksonomized ontology* synthesizes complementary roles of ontologies and folksonomies. In one direction, the knowledge systematically organized and formalized in ontologies is "folksonomized", i.e., the latent semantics from the folksonomic tissue is extracted and fused to ontologies. On the other, the folksonomized ontologies are explored to enhance operations involving tags, e.g., content indexation and discovery. The folksonomic data fused to an ontology will tune it up to contextualize inferences over the repository.

Our approach was validated by a tool we developed, which extracts tags from Delicious and Flickr, fusing them in the WordNet [12] ontology. WordNet is a lexical database of English, having a formalized thesaurus, which can be used as ontology. The resulting folksonomized ontology shows better results when applied to content discovery.

This paper is organized as follows. In Section 2 we discuss the basis of our work. We present our solution in Section 3 and the experimental results in Section 4. In Section 5 we confront our approach with related work and we conclude

[1] http://blog.flickr.net/en/2010/09/19/5000000000/

[2] http://blog.delicious.com/blog/2008/11/delicious-is-5.html

WebMedia'11: Proceedings of the 17th Brazilian Symposium on Multimedia and the Web. Full Papers.
October 3 -6, 2011, Florianópolis, SC, Brazil.
ISSN 2175-9642.
SBC - Brazilian Computer Society

and discuss the future work in Section 6.

2. FOLKSONOMIES, ONTOLOGIES AND SIMILARITY

In this section we summarize some related work which subsidized our research.

2.1 Folksonomies and Ontologies

In folksonomy-based systems, users can attach a set of tags to resources. These tags are not tied to any centralized vocabulary, so the users are free to create and combine tags. Some strengths of folksonomies are their easiness of use and the fact that they reflect the vocabulary of their users [10]. In a first glimpse, tagging can transmit the wrong idea of a poor classification system. However, thanks to its simplicity, users are producing millions of correlated tags. It is a shift from classical approaches – in which a restricted group of people formalize a set of concepts and relations – into a social approach – in which the concepts and their relations emerge from the collective tagging [17]. In order to perform a systematic folksonomy analysis, to subsidize the extraction of its potential semantics, researchers are proposing models to represent its key aspects. Gruber [4] models a folksonomy departing from its basic "tagging" element, defined as the following relation:

$$Tagging(object, tag, tagger, source) \quad (1)$$

In which *object* is the described resource, *tag* is the tag itself – a string containing a word or combined words –, *tagger* is the tag's author, and *source* is the folksonomy system, which allows to record the tag provenience (e.g., Delicious, Flickr etc.).

In order to formalize a folksonomy Mika [11] departs from a tripartite graph with hyperedges. There are three disjoint sets representing the vertices:

$$T = \{t_1, \ldots, t_k\}, U = \{u_1, \ldots, u_l\}, R = \{r_1, \ldots, r_m\} \quad (2)$$

In which the sets T, U and R correspond to tags, users and resources sets respectively.

A folksonomy system is a set of annotations A relating these three sets:

$$A \subseteq T \times U \times R \quad (3)$$

The folksonomy itself is a tripartite hypergraph:

$$H(T) = \langle V, E \rangle \quad (4)$$

In which $V = T \cup U \cup R$, and $E = \{\{t, u, r\} \mid (t, u, r) \in A\}$

The folksonomy analysis can be simplified and directed by reducing this tripartite hypergraph into three bipartite graphs: TU relating tags to users, UR relating users to resources and TR relating tags to resources [11]. A graph TT is a relevant extension of this model for representing relations between tags. It allows to represent the co-occurrence of tags. The same approach can be applied to the user and resource sets.

The Gruber's classical definition of ontology as "an explicit specification of a conceptualization" [3] synthesizes its key aspect as an intentionally systematized – or engineered [11] – specification. According to Shirky [17], contrasting to ontologies, in tag-based approaches the organization derives from an organic work. It is a shift from a binary categorization approach – in which a concept A "is" or "is not" part of a category B – to a probabilistic approach – in which a percentage of people relates A to B. Gruber [4], on the other hand, claims that folksonomies and ontologies should not be seen as opposite but rather as complementary, and he proposes a TagOntology – a common ontology for tagging. As we will present in this paper, we share Gruber's view of complementary roles, expanding the perspective to introduce a fusion (bidirectional) approach, in which folksonomies meet "classical" ontologies. Kim et al. [6] described three areas where the association of ontologies and folksonomies can improve the systems, namely: knowledge representation sophistication, facilitation of knowledge exchange and machine-processable. Moreover, this association can improve the tag query and disambiguation, visualization of tag clusters and tag suggestion to users [18].

2.2 Similarity and Information Content

One way to explore the semantics – formalized in ontologies and potential in folksonomies – involves matching and similarity. There are many applications, such as, ontology engineering, information integration and web query answering where matching operations play a central role [2]. When tags are compared, matching operations can be organized in two main broad categories: lexical/syntactic and semantic. Lexical/syntactic approaches are mainly based on the proximity of spelling words and their derivations (e.g., conjugations). One example of this category is the edit distance, as the popular approach proposed by Levenshtein [8].

To go beyond the spelling, semantic approaches relate words to a respective semantic representation – a concept. The matching is evaluated by analyzing semantic relationships among concepts, e.g., equivalence, generalization, specialization etc. This approach can lead to better search results or expand the opportunity for discovery, by finding and ranking similar or related results. It can also subsidize better recommendation systems for tag definition. In this context, ontologies are increasingly being adopted to formalize the semantics of concepts and their relationships.

A challenge in semantic matching is how to weight the relevance of relationships when similarities are confronted. Consider a practical example of a program looking for the concepts similar to `judge`. The output will be a set of concepts ranked according their similarity. Two possible similar concepts in the example could be `district attorney` or `child`. Like a `judge`, the former is an official functionary and the latter is a person. To rank them by similarity it is necessary to define which concept is more similar to `judge`.

In order to put this comparison in a context, let us consider a classical abstraction of an ontology as a graph, in which each vertex (node) is a concept and each edge is a relationship between two concepts. A comparison supported by this ontology considers that the compared terms are connected by a path. Figure 1 illustrates the three previous concepts as they appear in the WordNet ontology. In the figure, circles represent concepts and edges subsumption relationships – lower concepts specialize upper ones.

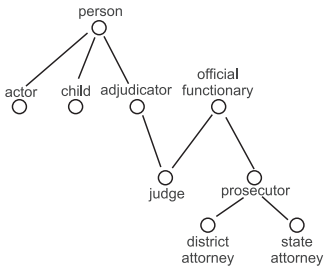

Figure 1: Subsumption ontology showing the relationships among compared concepts.

Many approaches to calculate semantic similarity based on ontologies were developed and we will further present some relevant techniques.

2.2.1 Path-based

A naive method to evaluate the semantic similarity between two nodes in an ontology is by measuring the shortest path separating them. This is equivalent to the distance metric in a *is-a* (subsumption) semantic net, defined by Rada et al. [13]: $distance(c_1, c_2)$ = minimum number of edges between c_1 and c_2. The similarity then calculated as:

$$sim_{rada} = [1 + distance(c_1, c_2)]^{-1} \quad (5)$$

As showed in [14], this approach is highly influenced by the level of detail applied to describe branches of the ontology, i.e., branches better detailed can contain longer paths than other, in spite of the similarity distance, leading to biased evaluations. For example, the comparison of `judge` with `child` (3 edges) results in the same similarity of `district attorney` compared to `judge` (3 edges). One way to overcome this limitation is by weighting the edges, leading to the problem of how to determine the weights. According to Jiang and Conrath [5] there are ontology aspects, such as depth of nodes and type of links, which can be used to define these weights.

2.2.2 Depth-relative

One way to enhance the path-based comparisons is by analyzing the most specialized common ancestor shared by two nodes in the ontology. It is founded in specific kinds of taxonomic ontologies based on subsumption relationships among terms, as the example of Figure 1. Observations showed that siblings sharing an ancestor deep in a hierarchy are more closely related than those sharing an ancestor higher in the hierarchy [19]. Therefore, Wu and Palmers [22] propose the following metric:

$$sim_{wp}(c_1, c_2) = \frac{2 \times N_3}{N_1 + N_2 + 2 \times N_3} \quad (6)$$

In which c_3 is the least (most specialized) common ancestor of both c_1 and c_2, N_1 is the number of nodes on the path from c_1 to c_3, N_2 the number of nodes between c_2 and c_3, and N_3 the number of nodes between c_3 and the ontology root.

To improve the depth-relative metrics, Shickel and Faltings [16] proposed the OSS metric, based on an A-Priori Score *APS* computation of all concepts in an ontology. Then, a distance metric is defined from two coefficients (generalization and specialization) calculated from the *APS* value.

2.2.3 Content-based

Besides the ontology topology, there are approaches showing that comparisons can be improved by analyzing also the content of the ontology concepts. Resnik proposed an approach based on *information content* [14] applied to subsumption ontologies. Assuming that each concept in this kind of ontology is a class representing a set of instances, the probability of a given instance to belong to a more specific class – e.g., `child` – is lower than the probability to belong to a more general one – e.g., `person`. While the probability decreases, the information about more specific classes increases – a necessary consequence of their specialization. *Information Content* (IC) is a measure created to evaluate this increase of information about something. Let the probability of a given concept `c` be `p(c)`, then the IC of `c` is `-log p(c)` [15].

In order to illustrate Resnik's IC-based approach to evaluate the similarity among terms, let us return to the example involving the similarity ranking among `judge` and two other concepts: `district attorney` and `child`. The first step is to find the most specialized concept shared by `judge` and `district attorney`, which is `official functionary`, as by `judge` and `child`, which is `person`. Intuitively, we can infer that the probability of an instance to belong to `official functionary` is smaller than the probability of an instance to belong to `people`; conversely the IC is higher. In this type of ontology, when two concepts derive from the same generalization they share its characteristics, therefore, `judge` is more similar to `district attorney` than to `child`, since the former has higher IC. Therefore, the Resnik [14] similarity metric was defined as follows:

$$sim_{res}(c_1, c_2) = \max_{c \in S(c_1, c_2)} [-\log p(c)] \quad (7)$$

In which the set S contains all concepts that subsume both c_1 and c_2. Experiments in [14] demonstrated that this approach produces better results than the counting edges approach and is not influenced by unbalances in ontology detailing. There are many other approaches exploring probabilities to improve similarity evaluation such as Lin [9] and Jiang and Conrath [5].

All of these probability-based approaches lead to an extra challenge: how to evaluate the probability of each concept of an ontology. Resnik's strategy is based on counting words extracted from a corpus of documents. As will be further detailed, our work expands Resnik proposal in three directions:
(i) proposing a strategy for calculating probabilities and IC of concepts based on tags employed in social networks to describe content;
(ii) defining multiple context-driven IC for each concept;
(iii) applying IC and co-occurrence data to review the ontology.

3. FOLKSONOMIZED ONTOLOGIES

As observed in the previous section, ontologies and folksonomies can play complementary roles. Nevertheless, existing proposals usually are unidirectional, attaching folksonomy's tags to ontologies or, conversely, producing on-

tologies from folksonomies. In this section we describe our fusion approach, which takes advantage of both ontologies and folksonomies, producing a synthesis. This fusion results in a *folksonomized ontology*, which we define as an ontology aligned with terms of a folksonomy and enriched with their contextual data. By contextual data we mean data which emerges from a statistical analysis of a folksonomy, e.g. tag frequency, co-occurrence and information content.

In one direction the folksonomized ontology, which is aligned with tags, drives richer semantic-based matching, categorization and tag suggestion. In the other direction, contextual data will be used to review and improve the ontology. The Figure 2 schematizes the roles played by an ontology and a folksonomy in a folksonomized ontology building. The ontology was previously engineered to formalize concepts and typed relationships, e.g., is-a, same-as, part-of. Concepts and relationships in folksonomies, on the other hand, are inferred by statistical analysis over tags and their co-relations. They are not typed, as in ontologies, but carry substantial contextual data, which subsidizes "weighting" concepts and relationships. The resulting folksonomized ontology is a new entity that fuses the best of both worlds, having typed and "weighted" concepts and relationships.

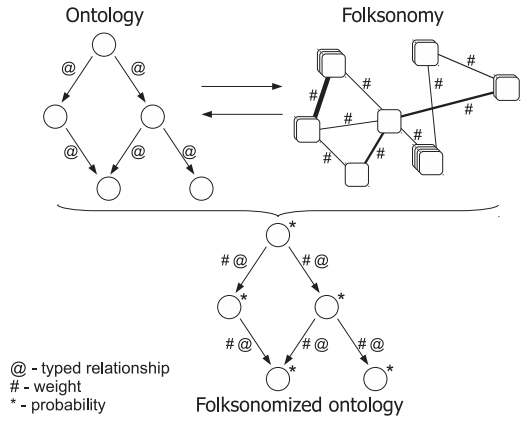

Figure 2: Folksonomized Ontology

A practical tool was developed in this research, apt to build folksonomized ontologies and use them for tag searching and discovery, as to ontology review and improvement. Figure 3 summarizes the cycle of the folksonomic ontology building and use. It starts collecting data from folksonomy systems (step 3.1), e.g., Delicious and Flickr, which are processed, filtered and grouped as concepts (concept-group) (step 3.2). Concept-groups are mapped to concepts in ontologies (step 3.3). The probability and IC for each concept-group, as the co-occurrence of concept-groups, are calculated and fused to the ontology, obtaining our folksonomized ontology (step 3.4). The step (3.5) is an ongoing work in this research; it confronts statistical data extracted from a folksonomy with the structure of an ontology, in order to subsidize ontology review and improvement.

The step (3.2) involves preprocessing algorithms, e.g., to adjust punctuation mismatches and to group tags. Since our contribution is not focused in these preprocessing algorithms, but rather in the subsequent steps, we implemented established algorithms, which will not be compared to related work. Moreover, we adopted the same preprocessing algorithms when comparing our approach to related work. In the following subsections each step of the process illustrated in Figure 3 will be detailed.

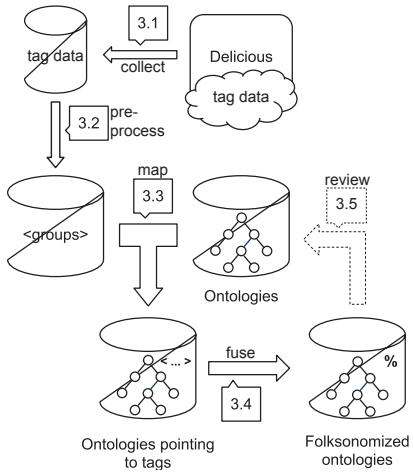

Figure 3: Folksonomized ontology building and use

3.1 Collecting Tag Data

Web-based content portals offer web service interfaces to access their data. The tag data collecting module (step 3.1) access these web services to select and retrieve tags and their metadata, which are stored in a database. Due to the heterogeneity in the APIs, this module was designed to be customizable and it was tested in Delicious and Flickr systems. To better obtain the emergent properties of the semantics extracted from folksonomies, this module was designed to afford large datasets. They are stored as triples of resources, users and tags, including their relations. Statistical data – e.g., co-occurrence between tags – were computed and stored during data collection, avoiding extra post-processing work. The updating process is incremental, i.e., it collects and stores just the differences of previous processings.

3.2 Tag Processing

In order to avoid the interference of wrong spelled tags or similar problems, unusual tags – with less than five occurrences – were eliminated to improve the quality of the data set. This procedure produces a collateral effect, since it also filters correct tags having a high IC value, due to their low-frequency. Therefore, we consider this a preliminary approach. In a future work, we intend to study the impact that low-frequency tags have in the results and if they should be kept or deleted.

The next step involves grouping tags referring to the same term. For instance, the tags *tip* and *tips* are tightly connected and represent the same term. The grouping algorithm is divided in two steps: (i) **punctuation analysis** – groups tags differing only in punctuation signs; (ii) **morphological analysis** – group tags by morphological relatedness.

A common approach in tagging systems is to delimit tags by spaces. In order to represent multiple word tags, users resort to different strategies, e.g., concatenating words with or without separating signs. By analyzing the similarity of tags without the punctuation we could group tags like *search-engine*, *search_engine*, and *searchengine*. These tags are clearly very close to each other and represent different user approaches for using multiple word tags. So, all punctuation signs of tags were removed, allowing to group tags that became equal without punctuation.

The morphological analysis and grouping go beyond spelling comparisons, considering morphological variations as singular and plural tags, or tags of different verb tenses. The algorithm retrieves morphological variations of tags from the WordNet ontology, grouping them together.

3.3 Mapping tags to ontology terms

The next step, of mapping tags to ontology concepts, is not a simple task, due to the lack of semantic information related to the tags. The tags cannot be directly mapped based on their words, since the same word can have multiple meanings in the ontology. In WordNet, for instance, a word can have multiple senses, called synsets, which are differentiated through identifiers combining the original word plus two affixes. The first one is a character that describes the synset type (namely noun, verb, adjective, or adverb) and the second one is a sequential number to differentiate each meaning. For instance, the synset *dog.n.01* represents a noun and it is one of the synsets for the word *dog*.

To find out which synset corresponds of each tag, we developed a technique that encompasses the relation of the WordNet synsets and tag co-occurrences, divided in three steps: (i) group key election; (ii) co-occurrence selection; (iii) group key mapping. They are further detailed.

Group key election. In the previous stage, tags referring to the same term were grouped. In this step, a "group key" is elected to represent each of these groups. Since all variations of a tag in each group are considered referring to the same term, it is necessary to select the most significant to represent the group. Since the WordNet will be the target of the tag mapping, it is also used in the group key election process. By analyzing morphological derivations of words in the WordNet, it is possible to determine which word is the root in each group. This tag is elected the group key. There are exceptional cases in which it is not possible to fetch a root word for a given group. We implemented a preliminary solution in which the first tag in the group is elected. We are planning to implement a better approach for exceptional cases as a future work.

Co-occurrence selection. In order to put tag keys in a context, they are linked to related tags having highest co-occurrence values. Considering a group containing n tags. For each tag t in this group, the selection algorithm initially fetches the h tags having highest co-occurrence with t. The result is a set of $n \times h$ co-occurrences. Then, the algorithm selects the s tags with the highest co-occurrence values in this resulting set.

Group key mapping. The last step involves mapping group keys to WordNet's synsets. Consider a tag t, a group key, to be mapped to a synset and a set C containing the tags having the highest co-occurrence related to t (obtained in the last step). Consider a group S containing all synset candidates for mapping. Our algorithm evaluates the distance of each synset s of S compared with t, in the following way: (i) the set C must have a minimum set of tags already mapped to synsets; this minimum is defined by a threshold constant *minmap*; (ii) the similarity of a given s is calculated by the sum of the distances of all c already mapped to s; (iii) since there is no IC data yet, a path-based similarity algorithm is applied. The synset s with the highest sum is the target of the mapping.

A tag group will only be processed if a minimum of the elements in the corresponding co-occurrence list had already been processed and mapped. Since the algorithm always selects a synset based on tags already mapped, it was necessary to create a starting set of tags manually mapped, to work as seeds. Algorithm 1 presents a pseudo-code of the tag mapping.

Algorithm 1 The algorithm to map group keys to synsets

Input: G: set of groups keys (tags)
Input: minmap: minimum co-occurrence mapping
Output: S: set of group keys (tags) mapped to synsets

```
1: S ← {}
2: while ∃ t in G | fit(t) do
3:    t ← choose(G)
4:    cooc_list ← getcooc(t)
5:    list ← {}
6:    for all synset s in synsets(t) do
7:       for all element e in cooc_list do
8:          include (s,sim(s, synmap(e), coocval(t, e))) in list
9:       end for
10:   end for
11:   S[t] ← max(list)
12:   remove t from G
13: end while
```

The functions used in Algorithm 1 are:
choose(G) Returns a tag t in G in which $fit(t) = true$.
fit(t) Returns true if the co-occurrence list related to the tag t has at least *minmap* elements already mapped.
getcooc(t) Returns the co-occurrence list for the tag t, having the highest co-occurrence values and already mapped to a synset.
synsets(t) Returns all possible synsets for a given tag.
synmap(t) Returns a synset already mapped to a tag.
coocval(t_1, t_2) Returns the co-occurrence value between t_1 and t_2.
sim(s_1, s_2, e) Calculates the path-based similarity between the two synsets (s_1 and s_2) multiplied by the co-occurrence value (e).
max(list) Returns the synset having the highest similarity value in the list.

In the best scenario this algorithm stops when it maps all tags. However, depending on the starting seeds and the *minmap* value, it is possible that it will not converge and the algorithm will stop in the absence of eligible tags to process. In this case, the result is a partial mapping set.

After this step, a subset of WordNet ontology is mapped to tags of the folksonomy. However, there are WordNet concepts that do not point to tags. They are classified here as *virtual nodes* and the ones that point to tags are the *real nodes*. For instance, the term `entity.n.01` is the root of the ontology and does not point to any group of tags, a *virtual node*.

3.4 Fusing

After the mapping process, it is possible to calculate the information content (IC) of each ontology concept. Our algorithm starts by setting frequency values collected from the folksonomy in the *real nodes*. Each node change reflects in every predecessor node. The frequency is calculated by using the occurrences of the mapped tags.

This strategy considers that when users associate tags to resources, they are also associating the respective generalizations. For instance, when a user tags a resource with the tag *"judge"*, he is implicitly tagging this resource with the tag *"person"*. Since each tag frequency reflects in its predecessors, it is necessary to avoid counting twice when the same resource is tagged by a user with tags having a subsumption relationship – e.g., *"judge"* and *"person"*. These frequencies subsidize the calculus of probability and IC for each node.

4. PRACTICAL EXPERIMENTS

Among our practical experiments, in this section we will focus the presentation on Delicious data, due to the nature of its resources – URL addresses – which are better suited to compare with related work, as shown in the evaluation section. For the experiment discussed in this section, we have collected and stored a total of 1,049,422 triples of resources, users and tags, including their relations.

4.1 Similarity Algorithm

After calculating the IC values, we implemented some similarity and distance metrics like sim_{lin} [9], $dist_{jiang}$ [5] and sim_{res} [14] to validate our proposal. Considering that sim_{res} is a basis algorithm and sim_{lin}, $dist_{jiang}$ variations over it, our focus here will be the sim_{res} implementation.

Since Resnik's similarity metric is relative, it requires at least three terms: one pivot and two other terms to be ranked. Let's consider the pivot *graphic* and the comparing terms *picture* and *freeware*. With the sim_{res} we obtained, as expected, that *picture* is more similar to *graphic* than *freeware*.

In order to evaluate our folksonomized ontology in similarity operations, we conducted two groups of comparisons further described: (i) folksonomized ontology versus ontology and co-occurrence; (ii) folksonomy versus document emergent semantics.

4.2 Ontology and co-occurrence

We developed a qualitative analysis in tag comparison by confronting our proposal with: the WordNet ontology without folksonomized data and using path-based similarity algorithms; just tags and their co-occurrence statistics.

To present our considerations, we selected three representative cases of compared tag pairs: *graphics* and *inspiration*; *war* and *conflict*; *bible* and *christian*.

The terms *graphics* and *inspiration* have a high co-occurrence (41% of the maximum co-occurrence value for *graphics*), but low similarity in the path-based algorithm, since the terms are relatively far in the ontology (too much edges). The similarity based on folksonomized ontologies was more accurate in this case, since it does not rely solely on the ontology topology.

In the case of *war* and *conflict*, there are no co-occurrence value, because *conflict* does not exist in the tags dataset. But it exists in the ontology as a *virtual node* and has a high similarity with the term *war*. This example shows that with our folksonomized ontology it is possible to find similar terms and suggest them to the users, even if they do not exist yet in the tag dataset – a feature that a standalone folksonomy is not able to offer.

The pair *bible* and *christian*, however, shows a situation in which the co-occurrence has better results than the folksonomized ontology. Even having a high co-occurrence value (there is no tag with more co-occurrence with *bible* than *christian*), any ontology-based comparison of similarity (folksonomized or not) will return zero. The reason is that in WordNet the only common parent of these two terms is *"entity"*, the root of the ontology, leading to a zero similarity. The folksonomy points to a strong relationship between the terms and it is a valuable information, which can be used to support the ontology review, as shown in Subsection 4.4.

4.3 Document emergent semantics

In order to evaluate the potential semantics extracted from folksonomies, this second group of comparisons confronts data extracted from tags with those extracted from web pages. Since our tags were extracted from Delicious, each tag is related to a web page address (URL). Our experiment fetched approximately 4,500 web pages pointed by Delicious tags. The analysis of the pages content adopted the same technique used by Resnik, i.e., counting the words in the corpus to compute the word frequency.

Besides the IC computed by using the word count of pages, the rest of the process adopted the same algorithms of our solution. The resulting enriched ontology was used to the comparisons of which results we further present a qualitative and a quantitative analysis. The terms sim_{tag} and sim_{wc} will be used to refer the Resnik's similarity algorithm applied to our folksonomized ontology and to the other enriched ontology respectively.

4.3.1 Qualitative analysis

In this analysis we compiled a list with 100 triples containing: a pivot and two comparing terms. For each triple we manually marked the term that we judged to be more similar to the pivot and then the sim_{wc} and sim_{tag} were applied to the list.

The result of this analysis is that both similarities had a rate of 90% of conformity compared to our judgment. Both similarity algorithms had equivalent behaviors, i.e., both differed of our judgment in the same triples. This result shows

that both approaches achieved a good conformity rate and indicates a possible tendency to be explored, that in many contexts the semantics extracted from tags describing pages can avoid the analysis of the whole pages. In this preliminary analysis the results were confronted with our judgment, but the validation process will address users in future work.

4.3.2 Quantitative analysis

In the quantitative analysis, the two ontologies were confronted in exhaustive comparisons. For either ontology, a routine compared each concept with all other concepts of the same ontology. Since the similarity algorithm requires a third pivot concept, the pivot was randomly chosen in the ontology. The same comparison was made in parallel in both ontologies and the results were compared. To minimize the randomic effect of the pivot, the same algorithm was ran 100 times. The average number of different results obtained in similarity comparisons corresponds to 0,02% of the total of triples analyzed. Therefore, we conclude that both approaches are equivalent.

One could argue the differential of evaluating tags compared to the classic approach based on documents word counting. Besides the previous mentioned conclusion, pointing to the observation that tags could produce equivalent semantic results with less effort, since they are more focused in relevant aspects, tags are also available in a wide range of content management systems, which do not have text documents to be analyzed. In Flickr, for instance, the resources are pictures, thus it is not possible to use the approach of counting words. The folksonomized ontology can be tailored to each context by switching the folksonomy. Therefore, it is possible to consider a folksonomized ontology for pictures, other for links and so on. The same approach can be used to customize ontologies to specific domains.

4.4 Supporting the Ontology Review

Departing from the observations of this research, we envisage that folksonomized ontologies can support the review and improvement of the ontologies used as foundations. This can lead to a symbiotic cycle, in which folksonomized ontologies help to improve the underlying ontologies which, in turn, will improve the results of the folksonomized ontology.

Figure 4 shows a graph generated by a tool we are developing to review ontologies. The nodes in the graph represent concepts in the ontology. Nodes connected by arrows represent relations by concepts explicit in the ontology. Nodes connected by edges without arrows – e.g., *bible.n.01* and *christian.n.01* – represent concepts in the ontology without formal explicit relationships, having a high co-occurrence in the folksonomy. The thickness of the edge is proportional to the intensity of the correlation. It can signalize a missing relevant relationship, to be considered in the ontology review. This is a preliminary result of an ongoing work. Many other inferred data can be presented to support ontologies review.

5. RELATED WORK

we selected four relevant works among the initiatives relating folksonomies to ontologies to compare with our approach.

Specia and Motta [18] aimed to build ontologies from folksonomy data. They first preprocessed the tags, eliminating

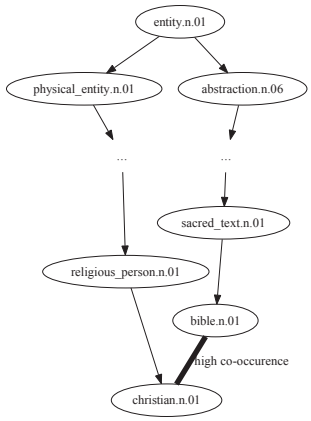

Figure 4: Example of folksonomy relationship absent in the ontology

the non-usual ones, and they created clusters of related tags, using co-occurrence information. Finally, they identified the relationships between these clusters using sources such as Google, Wikipedia and ontology bases. Damme et al. [20] proposed a system to group tags and associate them with ontologies. They used lexical resources, like Leo Dictionary, WordNet, Google and Wikipedia in the preprocessing step. A statistical analysis is applied to group tags in clusters.

Some steps followed by these works were followed by ours as well. The tag preprocessing, for instance, is a step that our work share with both. Our step of mapping tags into ontology terms differs from both. We focused in the folksonomy and ontology data, instead of looking for external sources. Different from both works, our approach takes fully advantage of the preexisting semantics in the underlying ontologies, instead of building a new ontology from scratch.

Konstas et al. [7] proposed a technique to filter tags, classifying them in categories, in order to infer the semantics of the classified tags to map them to knowledge bases like WordNet and Wikipedia. To find which category a given tag belongs, the authors resort to direct association or natural language processing heuristics. Cattuto et al. [1] applied existent ontologies, specifically WordNet, to find similarities between tags. However, their mapping approach do not group similar tags, resorting to a simple word comparison to find equivalent WordNet concepts. Our approach goes beyond, mapping groups of tags to synsets semantically related, even if syntactically they are not.

All of these related approaches are unidirectional, i.e., they produce ontologies from folksonomies or, conversely, use ontologies to assist tag relations in folksonomies. The major difference in our fusion approach is the symbiotic combination, in which ontologies support tag comparison and, on the other hand, folksonomies enrich (folksonomize) ontologies, improving their inferences and supporting ontologies review. In this sense, ontologies in our approach are not limited to be a tool to improve folksonomies. On the other hand, our approach will always require a preexisting ontology in the intended domain. Which can limit its application in some scenarios.

6. CONCLUSION AND FUTURE WORK

Folksonomy-based systems have been largely adopted on the web, due to their flexibility and easiness of use. However, these systems have limited search mechanisms, based on lexical comparisons of tags. On the other hand, formal categorizations, as ontologies, require a big effort to be built and maintained and do not take advantage of the potential semantics, which emerges in an organic way from social tagging systems.

To face this problem, this paper presents our approach to build a folksonomized ontology, an ontology fused with a folksonomy. It is a symbiotic combination, taking advantage of both semantic organizations. Ontologies provide a formal semantic basis, which is contextualized by folksonomic data, improving operations over tags based in ontologies. Conversely, the folksonomized ontologies can be also used as tools to analyze the ontology quality and to help the process of ontology evolution, showing the discrepancies between the emergent knowledge of a community and the formal representation of this knowledge in the ontology.

We are working to expand our research in the following directions: (i) to develop an interchangeable folksonomic dataset, providing different customizations of the ontology, according the context; (ii) to use other similarity algorithms and statistical data; (iii) to run tests in specialized contexts applying domain ontologies; (iv) to extend the solution to consider other relations in the ontology (besides the generalization and specialization); (v) to improve our tool for ontology evaluation and review; (vi) to measure and evaluate the costs and impact of our approach in current folksonomies.

Acknowledgments

This work was partially financed by CNPq, FAPESP, CAPES-COFECUB (AMIB project) and INCT in Web Science (CNPq 557.128/2009-9).

7. REFERENCES

[1] C. Cattuto, D. Benz, A. Hotho, and G. Stumme. Semantic grounding of tag relatedness in social bookmarking systems. In *The Semantic Web – ISWC 2008, Proc.Intl. Semantic Web Conference 2008*, volume 5318 of *LNAI*, pages 615–631, Heidelberg, 2008. Springer.

[2] J. Euzenat and P. Shvaiko. *Ontology Matching*. Springer-Verlag, 2007.

[3] T. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.

[4] T. Gruber. Ontology of folksonomy: A mash-up of apples and oranges. *International Journal on Semantic Web & Information Systems*, 3(2):1–11, 2007.

[5] J. Jiang and D. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the International Conference Research on Computational Linguistics*, Taiwan, 1997.

[6] H. L. Kim, S. Scerri, J. G. Breslin, S. Decker, and H. G. Kim. The state of the art in tag ontologies: a semantic model for tagging and folksonomies. In *DCMI '08: Proceedings of the 2008 International Conference on Dublin Core and Metadata Applications*, pages 128–137. Dublin Core Metadata Initiative, 2008.

[7] I. Konstas, J. M. Jose, and I. Cantador. Categorising social tags to improve folksonomy-based recommendations. *World Wide Web Internet And Web Information Systems*, 9(1):1–15, 2010.

[8] V. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966.

[9] D. Lin. An information-theoretic definition of similarity. In *Proceedings of the International Conference on Machine Learning*, pages 296–304. Morgan Kaufmann, San Francisco, CA, 1998.

[10] A. Mathes. Folksonomies - cooperative classification and communication through shared metadata. *Computer Mediated Communication*, 2004.

[11] P. Mika. Ontologies are us: A unified model of social networks and semantics. *Journal of Web Semantics*, 5(1):5–15, 2007.

[12] G. A. Miller. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995.

[13] R. Rada, H. Mili, E. Bicknell, and M. Blettner. Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man and Cybernetics*, 19(1):17–30, 1989.

[14] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In C. R. Perrault, editor, *Proceedings of the 14th IJCAI*, pages 448–453, Montréal (Canada), 1995.

[15] S. Ross. *A First course in probability*. Macmillan, 1976.

[16] V. Schickel-Zuber and B. Faltings. Inferring user's preferences using ontologies. In *proceedings of the 21st national conference on Artificial intelligence - Volume 2*, pages 1413–1418. AAAI Press, 2006.

[17] C. Shirky. Ontology is Overrated: Categories, Links, and Tags. `http://www.shirky.com/writings/ontology_overrated.html`, 2005. Retrieved on May, 2011.

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

[19] M. Sussna. Word sense disambiguation for free-text indexing using a massive semantic network. In *Proc. of 2nd International Conference on Information and Knowledge Management*, Arlington, Virginia, 1993.

[20] C. Van Damme, M. Hepp, and K. Siorpaes. 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"*. Springer, 2007.

[21] T. Vander Wal. Folksonomy. `http://vanderwal.net/folksonomy.html`, 2007. Retrieved on April, 2011.

[22] Z. Wu and M. Palmer. Verb semantics and lexical selection. In *Proc. of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138, 1994.