

Improving Search and Navigation by Combining Ontologies and Social Tags

Silvia Bindelli*, Claudio Criscione†, Carlo A. Curino*, Mauro L. Drago†,
Davide Eynard*, and Giorgio Orsi*

Dipartimento di Elettronica e Informazione, Politecnico di Milano,
via Ponzio 34/5, 20133 Milano, Italy

* < *surname* >@elet.polimi.it

† < *name.surname* >@gmail.com

Abstract. The Semantic Web has the ambitious goal of enabling complex autonomous applications to reason on a machine-processable version of the World Wide Web. This, however, would require a coordinated effort not easily achievable in practice. On the other hand, spontaneous communities, based on social tagging, recently achieved noticeable consensus and diffusion. The goal of the *TagOnto* system is to bridge between these two realities by automatically mapping (social) tags to more structured domain ontologies, thus, providing assistive, navigational features typical of the Semantic Web. These novel searching and navigational capabilities are complementary to more traditional search engine functionalities. The system, and its intuitive AJAX interface, are released and demonstrated on-line.

Key words: ontology, tag, web2.0, social bookmarking, search engine

1 Introduction

The Semantic Web is the “high road” toward a better exploitation of the vast amount of heterogeneous data available in the web. The overall goal is to mediate the access to existing sources, by means of formalized, shared, and explicit representation of the data semantics through ontologies, and to deliver value added interactions. This “high road”, appreciated in the academic environments, requires high switching costs and a wide distributed and coordinated effort, which is hard to achieve in practice. On the other hand, the recent phenomenon of the Social Web and in particular of tag-based systems represents a more practical and viable “low road” toward a better fruition of the web. The goal of the *TagOnto* system is to bridge the two roads, by automatically mapping tag-based systems with the more structured world of ontologies. The main contribution of our approach is to enhance the user experience by providing features typical of the “high road” while requiring only limited commitment, typical of the “low road”, from users and content providers. The system exploits a rich set of heuristics, ranging from simple string-distance measures to web-based tag disambiguation techniques, to discover correspondences between tags and concepts

of domain ontologies.

Therefore, the unstructured and uncontrolled nature of the *folksonomies*—as often the social tagging systems are named—is balanced by the formal rigor of the ontology-based component of our system. *TagOnto* enriches the user browsing experience by enhancing navigation and tag-based search with ontology-based search capability, which allows to disambiguate tags and to focus the user attention. The system platform is available for download and testable as an on-line demo¹. Both in the demo and in the paper we use the simple and well-known Wine ontology² as a running example. To show system extensibility we integrate in this example not only the standard tag engines such as *del.icio.us*, but also the wine community *Vinorati*.

The paper is organized as follows: Section 2 provides a brief overview of the system functionalities, Section 3 summarizes background knowledge, Section 4 discusses the internals of the system from a conceptual point of view, while architectural aspects are discussed in Section 5. Section 6 presents some related works, and Section 7 draws our conclusions.

2 System Overview

TagOnto is a *folksonomy aggregator* that offers services to relate, navigate and combine results of different tag-based systems. The key features of the system are: *a tag-based search engine*, mashing up several folksonomies to retrieve resources (bookmarks, images and videos); *an ontology-based query refinement*, exploiting a domain ontology, co-occurrence of tags and disambiguation techniques to filter prior results; and *an ontology-based navigation interface*, allowing the user to retrieve further results by graphical navigation of the ontology concepts. The above features provide two orthogonal and complementary ways, typical respectively of social and semantic web, to navigate the search results: associated-tag and ontology-navigation. The ontology is used as a common vocabulary and

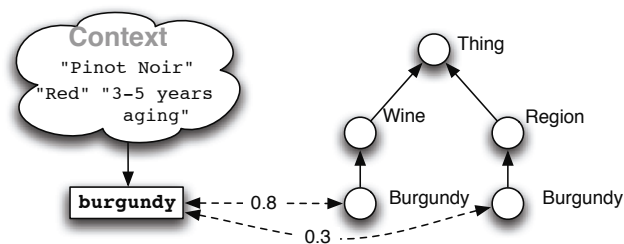


Fig. 1. An example of tag to ontology matching.

bridges the various folksonomies integrated in the system as a global schema

¹ The on-line demo can be reached from: <http://kid.dei.polimi.it/tagonto>.

² Available at: <http://www.w3.org/TR/2003/CR-owl-guide-20030818/wine>.

of a federated database; the system provides facilities to efficiently load the desired ontology before starting a web search. The typical user interaction is the following: (i) the user searches for a tag, e.g., **burgundy** (see Figure 1), (ii) navigates the concept of the associated ontology to refine the query, e.g., by selecting burgundy as a *wine* instead of as a *region*, or (iii) makes the query more general by navigating on more abstract concepts in the ontology. These actions are intuitively supported by the AJAX interface discussed in Section 5.

The associations between tags and ontology concepts are automatically discovered by the system, but also added, improved and maintained collaboratively. The automatic discovery of associations between folksonomies and domain ontology, represented as dashed lines in Figure 1, is based on a set of matching algorithms computing similarities. Disambiguation heuristics are then used in order to debug multiple associations between tags and concepts in the ontology.

Folksonomies are accessed by using dedicated wrappers exploiting three main methods to retrieve the needed resources: (i) Web2.0 APIs, (ii) RSS feeds, and (iii) Page scraping. The first approach, by relying on existing APIs offered by Web2.0-enabled websites, is our preferred one. In the second approach, the source of information is an RSS feed parsed and processed by a dedicated wrapper. The last technique is used when no other solutions are available; *TagOnto* uses page scraping to retrieve the needed information by making extensive use of regular expressions over the webpages to obtain tags and resources associated with them.

3 Background

We now introduce key background notions used throughout the paper:

Ontology: an ontology, according to T. Gruber [13], defines a set of representational primitives which can model a domain of knowledge or discourse. We can formally define an ontology as a 4-tuple $O = \langle \mathcal{C}, \mathcal{R}, \mathcal{I}, \mathcal{A} \rangle$ where \mathcal{C} is a set of *concepts* (or classes) which are subsets of a common domain Δ , \mathcal{R} is a set of relations including both binary relations between classes, called *roles* and binary relations between concepts and datatypes, called *attributes*, \mathcal{I} is a set of individuals (or ground symbols) which belong to Δ , and \mathcal{A} is a set of axioms (or assertions) in a logical form which are valid in the domain and restrict the number of possible interpretations of the ontological model.

Folksonomy: folksonomies are commonly defined as *the result of personal free tagging of information and resources for one's own retrieval* [17]. A *tag* in *TagOnto* is represented as a pair $T = \langle t, u \rangle$ where t is a term and u is a web resource (i.e., URL, image or video). The tagging is done in an open (social) environment, thus, the system is generated from the tagging performed by the people, which act both as tag providers and consumers. The term folksonomy derives from *folk* (people) and *taxonomy*. This is, however, often misleading since folksonomies lack the structure typical of taxonomies.

4 Matching and Disambiguation

As sketched in Section 1, one of the main problems in *TagOnto* is how to match a tag to a concept in the ontology. Given a tag and a reference domain ontology, the *matching process* (i) searches the ontology for *named concepts* whose name matches the tag, and (ii) looks for *related terms* which may refine the query for a better search. Moreover, (iii) a disambiguation process is often needed to reduce the noise produced by the collaborative tagging. Once the association has been created, the matched concepts are associated to each resource tagged by the corresponding tags. More precisely, given the set \mathcal{T} of all the available tags and the set \mathcal{C} of all the named concepts defined in a specific ontology, the matching is defined as a relation $M \subseteq \mathcal{T} \times \mathcal{C}$. The relation M allows multiple associations between tags and concepts. Figure 1 shows an example of such ambiguity: the term *Burgundy* might be referred either to the wine with that specific appellation or the region of France where that particular wine is produced. To distinguish the two different word acceptations, *TagOnto* associates to each matching a similarity degree by introducing the function $s : \mathcal{T} \times \mathcal{C} \rightarrow [0, 1]$.

To establish the matchings and to compute the similarity degree, *TagOnto* relies on the set of matching algorithms shown in Table 1. The matching algorithms can be classified on the basis of their effect on the set of matchings, in particular we distinguish between *generators* which generate new matchings starting from a tag and previous matchings and *filters* which choose the best candidates from a set of matchings. Another classification considers the metrics used to compute the matching degrees; we can distinguish between *language-based matching* which uses only morphological and lexical information such as string-distance metrics to compute the similarity and *semantic matching* which uses semantic and background knowledge to create new matchings. Notice that the matching problem has been extensively studied for ontologies [6] and many different classifications are present in the literature. In our context, the main difference is the absence of structure in folksonomies which does not allow an exploitation of structural similarities between the terms in the folksonomy and those in the ontology. Language-based generators use well known string-distance metrics, such as Jaccard similarity and Levenshtein distance. On the contrary, an example of language-based filter is the *Google Noise* algorithm, which suggests possible corrections for misspelled keywords by using the “did you mean”

	Language-based	Semantic
Generators	Levenshtein Distance Jaccard Similarity Google Noise Correction Concept Instances Similarity	Wordnet Similarity
Filters	Max Threshold	Graph Connectivity Neighbors Google Search

Table 1. Some matching heuristics.

feature of Google. In a similar way, a semantic generator is the *WordNet Similarity* algorithm which computes the Leacock-Chodorow [12] distance metric in WordNet between the term used in the tag and the concepts of the ontology. In *TagOnto* we use the implementation of the algorithm which is used in X-SOM (eXtensible Smart Ontology Mapper) [2] since it offers some extensions to handle compound words, acronyms and language conventions which are quite common in both folksonomies and ontologies. Since *TagOnto* is supposed to work online and with a fast response time, the class of syntactic filters includes some rather simple algorithms to select the best candidate matchings for a given tag, some examples are the *threshold filter*, which selects only matchings having a similarity degree greater than a specified threshold, and the *max filter* which selects the k matchings with the highest similarity degree. On the contrary, semantic filters are extremely useful in the disambiguation process since they alter the similarity degree of a matching by analyzing the concepts correlated to a tag using the structural information of the ontology. The disambiguation process is composed of two steps: (i) given a tag, the most frequent co-occurring tags are retrieved in order to specify its meaning (i.e., its *context*), and (ii) the ontology is analyzed in order to identify the concept which the closest meaning to the tag in that particular context.

The first process is carried out by the *Google filter* algorithm which retrieves the co-occurrent tags by issuing a query into Google and analyzing the first result. The second step, called *Neighbors filtering* leverages a common functionality of tag-based systems: the *tag-clouds*, which associate to each tag another set of tags whose meaning is correlated to the original one. After this information has been retrieved, *TagOnto* updates the similarity degrees of the matchings. As an example (see Figure 2) suppose we have the tag **Burgundy** with multiple matching concepts in the ontology (called *root concepts*); in first place *TagOnto* matches the co-occurrent tags obtained from tag clouds with the concepts of the ontology. The second step leverages the structure of the ontology by counting, for each matching, the number of links which connect matched concepts with each root concept, producing a vector of connectivity degrees v . The last step modifies the matching degrees of the root concepts according to the connectivity degrees computed in the previous step. For each matching i , *TagOnto* computes an offset measure $\varepsilon_i = \frac{D[i]}{MAX(v)}$ which is compared with the average connectivity $AVG(v)$; if $\varepsilon_i < AVG(v)$ then the new matching degree is decreased by a factor $\alpha \cdot \varepsilon_i$ where $\alpha \in [0, 1]$ is a configurable discount factor (currently set to 0.2 after the test phase); in the same way, the matching degree is increased if $\varepsilon_i > AVG(v)$. If the updated matching degree exceeds the values in $[0,1]$ the value is truncated to fit the range.

How these heuristics are combined depends on the selected matching strategy. We provide two different strategies: a *greedy strategy* which first invokes the syntactic and semantic generators and then applies the syntactic filters, and the *standard strategy* which invokes the greedy strategy and then disambiguates the results by invoking semantic filters. When tagging occurs in small communities of practice, which share a specific vocabulary without many ambiguities, the

greedy strategy can provide results comparable with the standard one, but in a shorter time. Whenever, instead, the user base is large and heterogenous such as on the Internet, the higher cost of semantic disambiguation is compensated with a much higher quality.

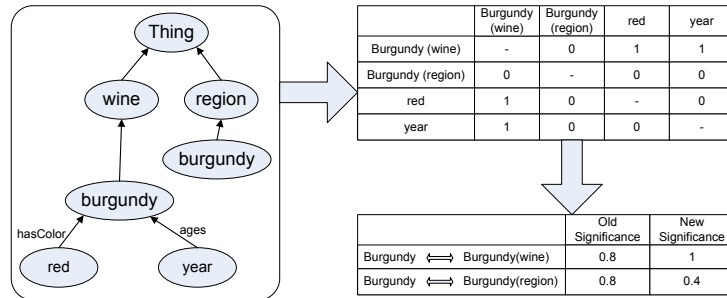


Fig. 2. An example of the disambiguation process.

5 Architecture

The overall architecture of *TagOnto* is logically divided into three different components: a tag-based search engine extensible with plugins, a heuristic matching discovery engine and a web-based user interface.

TagontoNET: TagontoNET provides core search engine functionalities and takes care of the integration of the results coming from folksonomies. The plugin-based architecture decouples the interaction between tag providers and *TagOnto*'s business logic. The system currently implements seven plugins to interact with some of the most popular tag-enabled websites such as Flickr, YouTube, del.icio.us, and Zvents. TagontoNET offers two main functionalities: tag-based resource retrieval and neighboring tag computation (needed by TagontoLib as discussed in the following). The results are delivered through a RESTful [7] web service, implemented in PHP, to further decouple this functionality, which might be used independently with the ontology-based portion of *TagOnto*.

TagontoLib: a Java library implementing the core matching functionalities of the system. The matching engine developed in Java implements the matching heuristics and strategies described in Section 4. To overcome performance limitations an effective caching technique has been built, maintaining recent matching tags and ontological concepts. As for the previous component, much attention has been devoted to the modularization of the tool. The communications between this library and the interface has been, in fact, based on a REST-like

communication paradigm [7].

TagOnto Web Interface: one of the distinguishing features of *TagOnto* is its web Interface which offers to the user the support of the Ontology within a comprehensive view of the results collected from a number of different tag engines. Users can import new ontologies into the system just by entering their URIs into a special page. The interface is then divided into two horizontal portions: the upper one reports the search results, the lower one is dedicated to the ontology representation and navigation. Each user query triggers searches in both the ontology and the tag-engines. The results from these two sources are respectively shown in the upper and in the lower part of the page. This provides a unified view of the ontological meaning of a tag and the available resources (tagged with that keyword). It is possible to exploit the support of the ontology to improve the search by navigating the ontology and thus triggering a query refinement procedure that will retrieve more specific resources based on the associated tags.

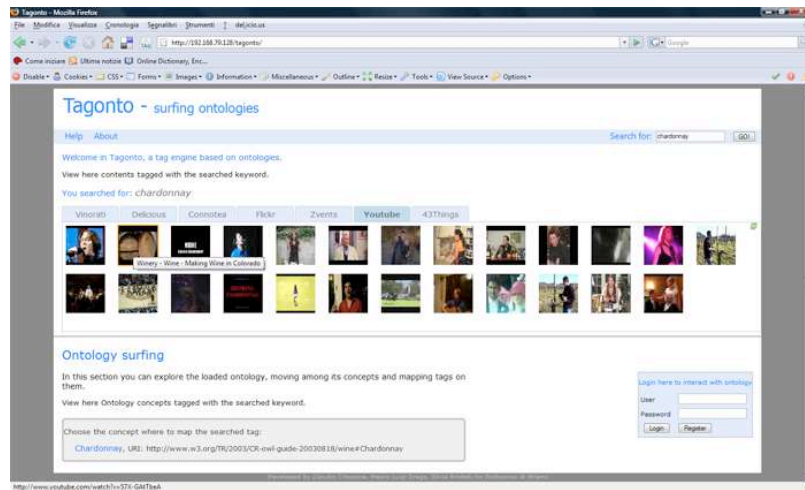


Fig. 3. The basic *TagOnto* web interface

The interface provides several tabs reporting the results obtained by searching each folksonomy. Textual results are presented in a “Google-like” way, while for picture results (e.g., Flickr resources) a thumbnail of the matching image is shown. The lower part of the page is dedicated to the presentation of the ontological concepts associated to the search. When a keyword is typed in the search field, a so-called “disambiguation box” appears in this area, to let the user choose among the concepts *TagOnto* computes as best matches. Once a concept has been chosen, previously mapped tags and resources are shown. The system also provides a box-based representation of other concepts related to the selected one, allowing an ontology-based navigation. During this navigation

process the co-occurrence of tags is used to provide feedback to the user and to suggest further directions for the exploration.

5.1 Performance

We measure system performance in terms of efficiency of the analysis and matching process, while an extensive usability study is part of our research agenda. To measure system efficiency, we stress test *TagOnto* when performing the two most expensive tasks occurring at run-time: (i) the time needed by *Tagonto* to analyze a new ontology to be deployed, and (ii) the time needed to automatically generate matchings. Figure 4 shows outcomes of our analysis. The time needed to perform an ontology analysis depends mostly on the number of concepts and properties declared in the ontology, with polynomial complexity as shown in Figure 4(a) while, with fixed concepts and properties (i.e., fixed schema), the number of instances declared in the ontology influences the execution time linearly as shown in Figure 4(b). Figure 4(c) shows the distribution of response time obtained by issuing 344 tag-queries (i.e., queries composed by a single term) taken from a set of terms referring to the wine domain.

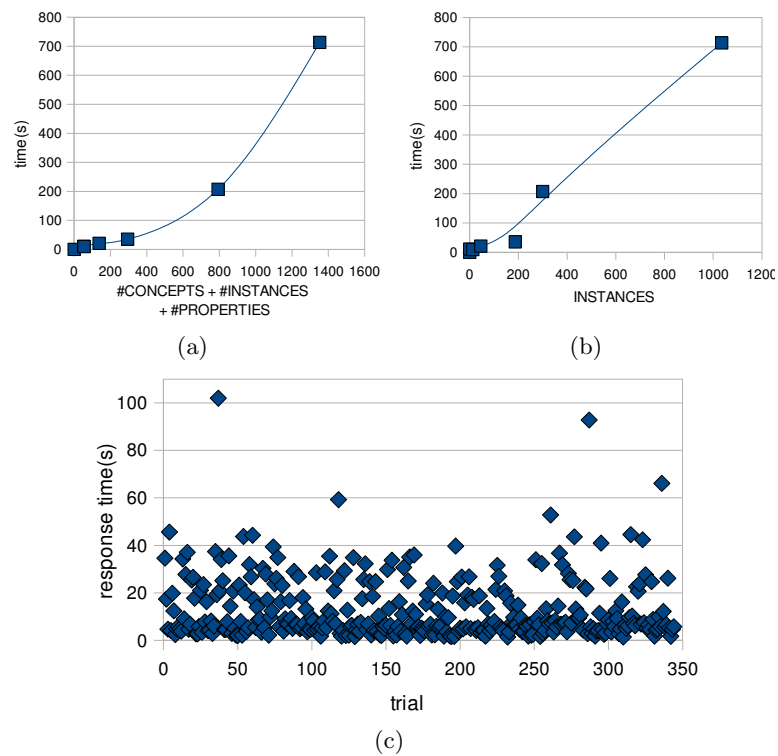


Fig. 4. Tagonto performance.

6 Related Work

The related works we consider can be grouped in (i) approaches which use ontologies to describe the domain knowledge and (ii) those which use ontologies to describe the tag system itself. SOBOLEO (SOcial BOokmarking and Lightweight Engineering of Ontologies, [18]) is a tool which allows to tag resources in the Web using ontology concepts and interacting with the ontology, modifying concept labels and relations. The SOBOLEO approach shares *TagOnto* objectives, but tries to exploit directly the ontology concepts as tags. [4] suggests an integrated approach to build ontologies from folksonomies, combining statistical analysis, online lexical resources, ontology matching, and community based consensus management. [1] presents an approach to enrich the tag space with semantic relations, “harvesting the Semantic Web”, by using the tool [16]. [3] addresses the problem of translating a folksonomy into a lightweight ontology in a corporate environment by exploiting the Levenshtein metric, co-occurrence, conditional probability, transitive reduction, and visualization. [15] uses the SIOC ontology in order to represent connections between tags and concepts of a domain ontology. [11] maps tags from del.icio.us with concepts from WordNet, and uses this mapping to provide an alternative interface for browsing tags. Gruber [9, 8] models the act of tagging as a quadruple (resource, tag, user, source/context) or a quintuple with a polarity argument, allowing to bind tagging data according to one particular system. Thus, tags from different systems can coexist in this model and it is possible to specify relations between them, allowing better interoperability. [14] defines a tag ontology to describe the tagging activity and the relationships between tags. [10] presents SCOT, an ontology for sharing and reusing tag data and for representing social relations among individuals. The ontology is linked to SIOC, FOAF and SKOS to link information respectively to resources, people and tags. [5] proposes a method to model folksonomies using ontologies. The model consists of an OWL ontology, capable of defining not only the main participants in the tagging activity, but also complex relations that describe tag variations (like `hasAltLabel` or `hasHiddenLabel`).

7 Conclusions

In this paper we presented *TagOnto*, a *folksonomy aggregator*, combining the collaborative nature of Web2.0 with the semantic features provided by ontologies, to improve the user experience in searching and browsing the web. The design of the system has been such that very limited overhead is imposed to users and content providers to enable these new features. *TagOnto* key components are a multi-folksonomy, tag-based search engine, and an ontology-based query refinement facility, which exploits a domain ontology to filter results and to focus users’ attention. In the best Web2.0 tradition, these features are delivered through an intuitive and reactive AJAX interface. The system is released and demonstrated online, and has been successfully tested on several domains. Nonetheless, we consider *TagOnto* a starting point for further developments and

we plan to devote more work on three key aspects: usability, performance and extensibility.

References

1. Sofia Angeletou, Marta Sabou, Lucia Specia, and Enrico Motta. Bridging the gap between folksonomies and the semantic web: An experience report. In *Bridging the Gap between Semantic Web and Web 2.0 (SemNet 2007)*, pages 30–43, 2007.
2. C. Curino, G. Orsi, and L. Tanca. X-som: A flexible ontology mapper. In *DEXA Int. Workshop on Semantic Web Architectures For Enterprises (SWAE)*, 2007.
3. Cline Van Damme, Tanguy Coenen, and Eddy Vandijck. Turning a corporate folksonomy into a lightweight corporate ontology. In *BIS*, volume 7 of *Lecture Notes in Business Information Processing*, pages 36–47. Springer, 2008.
4. Cline Van Damme, Martin Hepp, and Katharina Siorpaes. Folkontology: An integrated approach for turning folksonomies into ontologies. In *Bridging the Gap between Semantic Web and Web 2.0 (SemNet 2007)*, pages 57–70, 2007.
5. Francisco Echarte, Jos Javier Astrain, Alberto Crdoba, and Jess E. Villadangos. Ontology of folksonomy: A new modelling method. In *SAAKM*, volume 289 of *CEUR Workshop Proceedings*, 2007.
6. Jérôme Euzenat and Pavel Shvaiko. *Ontology matching*. Springer-Verlag, Heidelberg (DE), 2007.
7. Roy T. Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine, 2000.
8. Tom Gruber. Ontology of folksonomy: A mash-up of apples and oranges, 2005. <http://tomgruber.org/writing/ontology-of-folksonomy.htm>.
9. Tom Gruber. Tagontology - a way to agree on the semantics of tagging data, 2005. <http://tomgruber.org/writing/tagontology-tagcamp-talk.pdf>.
10. Hak Lae Kim, John G. Breslin, Sung-Kwon Yang, and Hong-Gee Kim. Social semantic cloud of tag: Semantic model for social tagging. In *KES-AMSTA*, volume 4953 of *Lecture Notes in Computer Science*, pages 83–92. Springer, 2008.
11. David Laniado, Davide Eynard, and Marco Colombetti. Using wordnet to turn a folksonomy into a hierarchy of concepts. In *Semantic Web Application and Perspectives - Fourth Italian Semantic Web Workshop*, pages 192–201, Dec 2007.
12. C. Leacock, M Chodorow, and G. A. Miller. Using corpus statistics and wordnet relations for sense identification. *Computational Linguistics*, 24(1):147–165, 1998.
13. L. Liu and M. T. Oszu. *Encyclopedia of Database Systems*. Springer, 2008.
14. R Newmann. Tag ontology design, 2005. <http://www.holygoat.co.uk/projects/tags/>.
15. Alexandre Passant. Using Ontologies to Strengthen Folksonomies and Enrich Information Retrieval in Weblogs. In *Proceedings of the First International Conference on Weblogs and Social Media (ICWSM)*, Boulder, Colorado, March 2007.
16. Marta Sabou, Mathieu d’Aquin, and Enrico Motta. Using the semantic web as background knowledge for ontology mapping. In *Ontology Matching*, volume 225 of *CEUR Workshop Proceedings*, 2006.
17. T. Wander Wal. Definition of folksonomy. In *Online blog post at http://www.vanderwal.net/folksonomy.html*, 2004.
18. Valentin Zacharias and Simone Braun. Soboleo – social bookmarking and lightweight engineering of ontologies. In *CKC*, volume 273 of *CEUR Workshop Proceedings*, 2007.