Bridging the Gap between Semantic Web and Web 2.0 (SemNet 2007)

International Workshop at
the 4th European Semantic Web Conference
in Innsbruck, Austria, June 7, 2007.

Bridging the Gap between Semantic Web and Web 2.0 (SemNet 2007)

New kinds of highly popular user-centered applications such as blogs, folksonomies, and wikis, have come to be known as "Web 2.0". The reason for their immediate success is the fact that no specific skills are needed for participating. These new kinds of tools do not only provide data, but also generate a lot of weakly structured meta data. One perfect example is tagging. Here users add tags to resources, which can be seen as a kind of meta data. Tags are supposed to describe resources from the user's point of view. Such meta data is easy to produce, but it lacks any kind of formal grounding such as used in the Semantic Web.

On the other hand, the Semantic Web complements the described bottom-up effort of the Web 2.0 community in a top down manner as one of its central points is a fixed vocabulary, typed relations, and a more formal knowledge representation based on some kind of ontology. Users will typically have some conceptualization in mind when they provide their information, but for the researcher this is hidden in the data and needs to be extracted to be useful. Techniques to analyze network structures or weak knowledge representations like those found in the Web 2.0 have a long tradition in different other disciplines such as social network analysis, machine learning, or data mining. These kinds of automatic mechanisms are necessary to extract the hidden information and to reveal the structure in a way that the Semantic Web community can benefit from, and thus provide added value to the end user. On the other hand the established way to represent knowledge gained from the unstructured data can be beneficial for the Web 2.0 in that it provides Web 2.0 users with enhanced Semantic Web features to structure their data.

The aim of this workshop is to bridge the gap between the Semantic Web and the Web 2.0 communities. Since both communities work on graph-structured data, analysis methods from fields like social network analysis, graph theory, machine learning, or data mining could form a link between those communities. By bringing together researchers from different fields, we aim to achieve this goal.

For this workshop we had 27 submissions of which we accepted 12. These papers can be divided into two categories: position papers (4 papers, 8 pages each) and full papers (8 papers, 12 pages each). The papers cover the full range of Web 2.0 applications from tagging to wikis and propose the combination with Semantic Web approaches. Some approaches integrate Semantic Web directly into Web 2.0 applications. Other contributions borrow methods from social network analysis, machine learning, and data mining to extract patterns from Web 2.0 data which then are connected to Semantic Web data.

One example is the system presented by Abbasi et al., which provides a mechanism to organize resources by classifying the tags (or keywords) attached

to them into predefined categories. They propose a new classification algorithm that does not require training data. Angeletou et al. believe that content retrieval can be improved by making the relations between tags explicit. They propose the semantic enrichment of folksonomy tags with explicit relations by harvesting the Semantic Web, i. e., dynamically selecting and combining relevant bits of knowledge from online ontologies. Basile et al. propose a smart tag recommender able to learn from past user interaction as well as the content of the resources to annotate. Brandes and Lerner analyze a network that arises from the fact that Wikipedia articles undergo recurring editing. The goal of their paper is to assess the meaningfulness of the co-revision network. Ding et al. propose in their position paper an active semantic space as a showcase. Such a space is built on a combination of Semantic Web, service agent, and Web 2.0 technologies. Heath et al. present a methodology and algorithms that, by exploiting existing Semantic Web and Web 2.0 data sources, help individuals to identify who knows what in a social network, and who is the most trustworthy source of information on each particular topic. Identification of the most trustworthy sources is enabled by a rich trust model of information and recommendation seeking in social networks. Herzog et al. examine and discuss the properties of metadata in social systems (folksonomies) compared to metadata in semantic systems (ontologies). They present an idea for creating a link between a folksonomy and an ontology in order to combine the usability and flexibility of folksonomies with the precision of ontologies for a semantic search application.

Lange argues that current semantic wikis lack scientific services because domain-specific ontologies are not properly integrated. Thus he proposes the basic architecture of a semantic wiki centered around an ontology of scientific markup languages. Mitschick et al. address conceptual and technical issues of Web search within community-built Semantic Web content to retrieve useful information for personal media annotation. Siorpaes and Hepp propose the use of wiki technology in order to enable collaborative and community-driven ontology building. Thus, users with no or little expertise in ontology engineering are given the opportunity to contribute. Szomszor et al. investigate the integration of a movie folksonomy with a semantic knowledge base about users and movie rentals. Van Damme et al. argue that the social interaction manifested in folksonomies and in their usage should be exploited for building and maintaining ontologies. They then sketch a comprehensive approach for deriving ontologies from folksonomies by integrating multiple resources and techniques. Overall these papers show a wide range of contribution from more theoretical to experimental. We believe that they built a very good basis for discussion and further research in this emerging field of bridging the gap between Semantic Web and Web 2.0.

We thank the members of our program committee for their efforts to ensure the quality of accepted papers. We are looking forward to interesting presentations and fruitful discussions.

May 2007

The SemNet 2007 team
Bettina Hoser and Andreas Hotho.

Workshop Chairs

Bettina Hoser
Institute for Information Systems and Management
Universität Karlsruhe (TH), Germany
http://www.em.uni-karlsruhe.de

Andreas Hotho
KDE Group at University of Kassel
D–34121 Kassel, Germany
http://www.kde.cs.uni-kassel.de/hotho

Conference Organization

Programme Chairs

Bettina Hoser
Andreas Hotho

Programme Committee

Harith Alani
Bettina Berendt
Ulrik Brandes
Ciro Cattuto
Laura Dietz
Yihong Ding
Scott Golder
Susanne Hoche
Nicholas John Kings
Peter Mika
Claudia Mueller
Marta Sabou
Harald Sack
Christoph Schmitz
Sergej Sizov
Avare Stewart
Gerd Stumme
Max Völkel
Markus Weimer
David Wood
Michael Wurst

Table of Contents

Searching Community-built Semantic Web Resources to Support Personal Media Annotation

Annett Mitschick, Ronny Winkler, and Klaus Meißner

Dresden University of Technology, Department of Computer Science
Chair of Multimedia Technology, 01062 Dresden, Germany
{annett.mitschick, ronny.winkler, klaus.meissner}@inf.tu-dresden.de

Abstract. Appropriate annotation of documents is a central aspect of efficient media management and retrieval. As ontology-based description of documents and facts enables exchange and reuse of metadata among communities and across applications, the annotation of personal media collections using Semantic Web technologies benefits from existing (and evolving) information sources on the Internet. This paper addresses conceptual and technical issues of Web search within community-built Semantic Web content to retrieve useful information for personal media annotation. After analyzing application scenarios, we introduce a generic and extensible Semantic Web Search Component, which facilitates specific search configurations. As a sample application, we deployed the component within our ontology-based media management system, including evaluation and remarks on quantity and quality of search results with regard to community-built Semantic Web content.

1 Introduction

Personal media collections comprehend knowledge representing context and individual view of the owner. This knowledge is the ultimate key for managing digital media collections in a way that is suitable for human beings. However, to enable applications to process and visualize navigation paths and arrangements based on people's knowledge, appropriate machine-processable descriptions are needed. Semantic Web technologies [1], [2] provide opportunities to create and share such ontology-based descriptions in a standardized way.

The question which arises is how those semantic descriptions could be generated or reused from existing information sources. To some extend information about digital documents exists explicitly in the form of annotations and metadata (for specific formats quite comprehensive and according to established standards like ID3, EXIF, IPTC, XMP, etc.). On the other hand, a substantial portion of knowledge results implicitly from the content itself (e.g. persons or locations depicted in a photograph), the structure and characteristic features of a document, etc. Machine Learning techniques and classification might solve the one or the other issue. However, the safest way to acquire semantic information is

to ask the user to make contributions. With regard to the user's comfort, manual annotation should be limited to the most necessary, reusing existing information on the local desktop or even on the World Wide Web.

In this paper we propose an approach to enhance annotation of multimedia documents using semantic resource descriptions and ontology models found on the WWW. The first part of Section 2 illustrates background and state-of-the-art of ontology-based media management and media annotation, referencing relevant related work. The second part addresses opportunities to search the Semantic Web (in particular RDF-based user/community generated content) using Web Services and crawler implementations. We also present a selection of use cases of media annotation supported by Semantic Web search. Section 3 introduces our Semantic Web Search Component, illustrated and evaluated in Section 4 using a sample application. Finally, conclusion and outline of future work is given in Section 5.

2 Ontology-based Annotation of Documents

Documents are or can be enriched with metadata and annotations in several ways and on several levels. Chakravarthy et. al. [3] introduced five "dimensions" of information associated with documents: *resource metadata* (e.g. creation date, author, etc.), *content annotation* (describing information within the document), *immutable knowledge* (e.g. knowledge from dictionaries), *informal knowledge* (e.g. knowledge not explicitly mentioned within the document), and *folksonomies* (cf. Flickr[1] or Del.icio.us[2]). Following this classification, a couple of solutions and applications for document annotation address the one or the other "level", depending on the used ontologies and concepts. However, most of the work on ontology-based annotation (like CREAM [4], AKTive Media [3]) proceeds from the assumption that, before annotation, an appropriate ontology has to be created or assigned as a description schema (top-down approach). If this is left to the user, modality and sense of annotations depend on his/her intension, which is even more difficult for non-ontology engineers.

Even if a lot of projects are dedicated to general "cross-media" annotation, regarding supported application scenarios they either focus more on text resp. Web content annotation (like Annotea [5]) or multi-media annotation (like M-OntoMat-Annotizer [6]). In order to ease the manual effort of annotation, several projects apply Information Extraction and Machine Learning techniques to populate descriptions (in particular Natural Language Processing in case of text documents). However, as annotations can hardly be automated completely (regarding subjective information, fuzzy knowledge, etc.) the user should be encouraged to make individual contributions. In this regard, aspects of community contributions and "social annotation" offer interesting opportunities. Bookmark and tagging services enjoy growing popularity as their success particularly bases on the low entry barrier [7].

[1] `http://www.flickr.com`
[2] `http://del.icio.us`

Ontology-based annotation of private media collections could profit from recent developments, not only restricted to the incorporation of folksonomies, but in general through information sources from the current Semantic Web, which will likely evolve with the help of the Web community and appropriate applications.

2.1 Finding Semantic Web Content on the WWW

Dedicated Semantic Web search engines facilitate a focused access to Semantic Web content. Their operating mode is similar to traditional Web search engines. Thus, a Semantic Web search engine also consists of a crawler ("robot" or "spider"), a database, and a search interface.

Crawling the Semantic Web is comparable to crawling the Web of HTML content [8]. A crawler starts with some seed URLs, downloads the corresponding documents, analyzes each document to gather further URLs for crawling and does context specific processing of the retrieved contents, like creating the searchable entries in the database. The last steps are repeated until a stop criterion is met (e.g. no more URLs to crawl, reached a predefined link depth, or gathered a predefined amount of documents). In case of a Semantic Web crawler the documents of relevance are those containing RDF-based data, and the goal of discovering unvisited URLs from previously retrieved RDF data can be achieved through evaluating statements with predicates which are capable of expressing relationships between documents, like *rdfs:seeAlso* or *owl:imports*.

There are several standalone Semantic Web crawler implementations, which can be used for purpose-built search engines or software projects. Some to mention here are the crawler of the KAON framework [9], the Slug crawler [10], and RDF-Scutter[3]. However, using a standalone crawler, exhaustive crawling is needed to create a passably extensive database of Semantic Web data. This requires considerable amounts of time, disk space, and Web transfers for collecting and maintaining the data. Therefore, available search services like Swoogle [11], which offers support for software agents via a REST interface [12], can be used more easily in an application to profit from rich databases. Currently, Swoogle has parsed and indexed more than 370 million triples from about two million Semantic Web documents[4]. It allows search for terms, documents, and ontologies (i.e. a subset of Semantic Web documents where the fraction of defined classes or properties is significantly higher than the fraction of instances). A Swoogle query is basically a set of keywords which should be found in the literal descriptions of indexed documents, terms, or in the URIrefs of defined classes or properties. A query initiated by a software agent is responded with an RDF/XML file containing the ranked search results. Testing Swoogle showed that its strength is more on the side of finding ontologies, than of finding documents with instance data. Nevertheless, the major drawback of using a remote search engine within applications is of course the dependency on its availability and maintenance, which should be taken into account.

[3] `http://search.cpan.org/src/KJETILK/RDF-Scutter-0.1/README`
[4] `http://swoogle.umbc.edu/index.php?option=com_swoogle_stats`

2.2 Using Semantic Web Content for the Annotation of Personal Media Collections

A variety of media analyzing and information extraction tools (e.g. [6]) are able to perform the task of extracting characteristic attributes and features (including inherent metadata) from media documents for the generation of semantic descriptions. As already mentioned, automatically extracted information might not be sufficient enough for an appropriate description. In the following, we give some conceivable use cases for the further refinement of basic, automatically generated information with the help of external resources, in particular retrieved by the Semantic Web search component, introduced in Section 3:

Assigning terms or categories from a glossary or thesaurus: The user wants to add a tag to a document to assign it to a category or concept. Actually, he is not sure about the proper term and wants to use existing definitions (perhaps a controlled vocabulary). He discovers a domain thesaurus on the Web (e.g. a SKOS [13] based document), which contains suitable items to assign to the document.

Referring to domain-specific descriptions of people, social events, communities, projects, etc.: Analyzing components may extract - among others - the name (family and given name) of the photographer from the metadata of an image. A resource description of this person was generated but without further information than the name literals. Searching the Semantic Web possibly returns a *Friend-of-a-Friend* (FOAF) or vCard description of the person (or one with a similar name). The user can decide to add the found resources to his model to extend the description of the photographer. Moreover, some documents might be related to resources, like events (e.g. a party, workshop, trip, etc.) or work projects. In addition to his personal view and context, the user might want to link to external descriptions maintained by a community.

Referring to a Web page with embedded RDF: Besides Semantic Web documents containing pure RDF resp. OWL data, RDFa [14] annotated XHTML documents could as well provide relevant resource descriptions, e.g. published events, contacts, etc. In addition to the previous use case, the user might want to keep the link to the annotated Web page containing the retrieved information.

Adopting domain specific description schemes: The basic ontology model might not be sufficient enough to describe special issues, regarding diverse interests, profession, and background of users (e.g. detailed interest in wine, zoology, classical music, etc.). A keyword-based search might lead the user to an appropriate ontology on the Internet which he could adopt.

Improve information extraction from text documents: The results of *Named-Entity-Recognition* (NER) in text documents could be qualified by semantic search results, i.e. tagging person names, addresses, locations, events, etc. within the document depending on found entities on the Semantic Web.

According to these and other identified application scenarios, we finally derived the following concepts of information reuse within the context of annotation, each with increasing complexity:

Tagging: Assigning tags to multimedia content (or generally any resource in the model) is probably the easiest way of information integration and does not necessarily require substantial adjustment of the ontology model. The RDF vocabulary [15] provides built-in utility properties for linking between general resources. One of those is *rdf:seeAlso*, which could be used as a simple tag relation between two *rdfs:Resource* instances. A better representation of the semantics of tagging might certainly be the definition of a tagging vocabulary (*hasTag*, *taggedBy*, etc.) to combine benefits of a controlled vocabulary with those of social tagging.

Referencing external objects: Found resources on the Web might be integrated as objects of a defined property if they fit in the required range, i.e. the same class or subclass. The practical application of this option depends on the constraints within the used ontology model. Proprietary object types of course complicate the creation of semantic nets to external resources.

Instance mapping: In the case of instance mapping, attributes and data of the retrieved resource are "translated" to slots of the target resource. Therefore, adjustment of the ontology model is not needed. Hints how to solve concrete mapping problems should be given by the user.

Refinement (specialization): A specialization of classes within the ontology model using retrieved class definitions or class definitions of retrieved instances might be useful. In the concrete application scenario the user introduces this subclass relation with a retrieved instance. He wants the target instance to adopt its properties, but keep the existing class definition unaffected. The retrieved class is incorporated into the ontology model as a copy and defined as subclass of the target class. The target instance is altered to an instance of the new class. Thus, existing relations to the instance are still valid.

Instance and schema adoption: The most complex scenario of information reuse from retrieved resources is the extension of the ontology model with both instances and their according schema. The user wants to incorporate a resource as object of a newly defined property of an existing resource. Thus, the ontology model has to be extended with the new property and a local copy of the adopted class.

Please note, that all of these concepts refer to crawled data in general, which could be downloaded from the Internet to local disk or used without local caching. Thus, as models, once retrieved, could change or get lost, cached data becomes obsolete, but without caching statements might become invalid. Therefore, its left to the developer to find a reasonable compromise.

3 Semantic Web Search Component (SWSC)

Based on the study of existing Semantic Web search solutions and use cases (see Sections 2.1 and 2.2) we developed a Semantic Web Search Component (SWSC), as depicted in Fig. 1. The SWSC is designed to extend applications of Semantic Web technologies with search functionality, including search for ontologies, documents with instance data, and terms. Instead of creating our own crawling infrastructure, we decided to reuse existing Web search services in the form of meta-crawling. As it seemed advisable to reduce the dependency on a single service, we provide an extensible meta-crawler concept (cf. Fig. 1), facilitating dedicated *Crawler* implementations handled by a central *CrawlerManager*. The main idea of this approach is a generic interface (*WebSearchInterface*) which accepts search requests and forwards them to the registered crawler implementations. A more detailed description of the search requests is given below in Section 3.1.

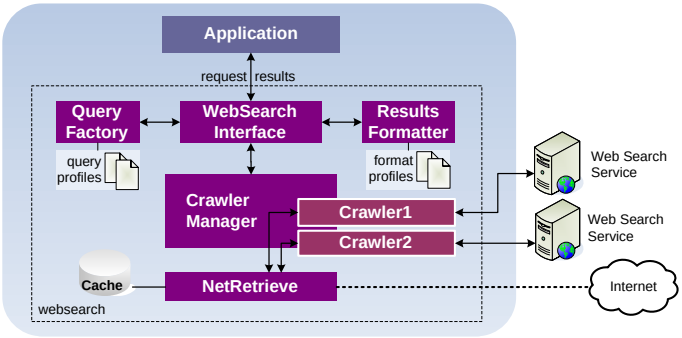


Fig. 1. Search concept and integration of the SWSC.

After processing the search task, each crawler implementation produces an initial result set of potential document URIs found on the WWW (i.e. indexed by the inquired Web service), which are evaluated in the following according to the given search criteria. To achieve the required Web communication, the *NetRetrieve*-component, offering multi-threaded downloads with local caching capabilities, was implemented. The *CrawlerManager* removes duplicate results, if documents have been found by several *Crawlers*, and applies the predefined filters of the *Query* object to the result set. The *ResultsFormatter* finally generates an adequate representation of the search results to be returned, according to specified "format profiles"(i.e. templates for XML, XHTML, or RDF response).

Trying to harvest community-built Semantic Web content, we decided to combine a dedicated and a general purpose search engine to achieve a better coverage, regarding the identification of potential instance data. Currently our SWSC implementation makes use of the publicly available Web interface of the

Semantic Web search engine Swoogle in combination with the general purpose Web search service of Yahoo!. As a matter of course, Swoogle's strength is the dedicated and exclusive access to Semantic Web content (ontologies, documents, and terms) which has already been evaluated and ranked. As mentioned before in Section 2.1, Swoogle's ability to supply instance data is relatively limited. Although the coverage of Yahoo! is estimated to be smaller than that of Google[5], we decided to work with Yahoo! as Swoogle itself already applies Google-based meta-crawling for its index [16].

3.1 Defining Search Requests and Results Filtering

Needless to say, requests to the Web interfaces of the search engines have to conform to the required query syntax and parameters. The implemented *Crawlers* serve as wrappers for the request specification and reception of results from the according Web interface. A general *Query* object is used to retain the implementation-independent query parameters and filter definition. The query string itself consists of keywords to be searched for. Additionally, different kinds of filters could be attached to a query to constrain the search. All these filters can be operated as blacklists or whitelists, allowing conjunction or disjunction. Currently, we use a host filter to include or exclude results from specific hosts, as well as a file type filter to restrict the result set of documents, a namespace filter which can be used to test if an RDF-model relies on a specific vocabulary, and a filter that checks terms whether they fulfill certain criteria (if they are class or property, or subject/objects of a specific statement). In general, the filters are applied in this order. Although not all of them can be mapped directly to the query syntax of the Web search interfaces[6], they are used to evaluate the retrieved documents locally to determine in more detail whether they match the query.

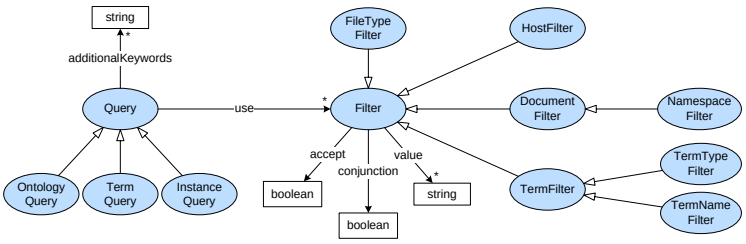


Fig. 2. Query profile ontology. For each filter (combined by conjunction) a set of restricting values (e.g. URLs) can be specified and combined by disjunction or conjunction (*disjunction=true/false*), and as whitelists or blacklists (*accept=true/false*).

[5] `http://blog.searchenginewatch.com/blog/050517-075657`, May 2005

[6] e.g. using file type restriction within the Web query with *url:* (Swoogle) resp. *originurlextension:* (Yahoo!)

Based on the requirements for a general query definition we created a purpose-built ontology for the Web query specification (an extract is given in Fig. 2). Thus, we are able to instantiate some sort of "query profiles" as instances for specific application scenarios (people search, schema search, etc.), which can be loaded by the *QueryFactory* to instantiate the appropriate *Query* object.

To prove applicability of Semantic Web search within the context of personal media annotation, we integrated the SWSC into our ontology-based media management system, which we illustrate in the following.

4 Semantic Web Search within the *K-IMM* Ontology-based Media Management System

This work is based on the results and implementation within the *K-IMM* (Knowledge through Intelligent Media Management) project, which provides a system architecture for intelligent media management for private users (i.e. semi- or non-professionals) [17]. The intention of this project is to take advantage of ontology engineering and Semantic Web technologies in such a way that users without particular skills can interact intuitively and without additional cost. Therefore, the system comprises components for automated import and indexing of media items (of different type) as background tasks [18]. A conceptual overview of the overall architecture of the *K-IMM* System is depicted in Figure 3.

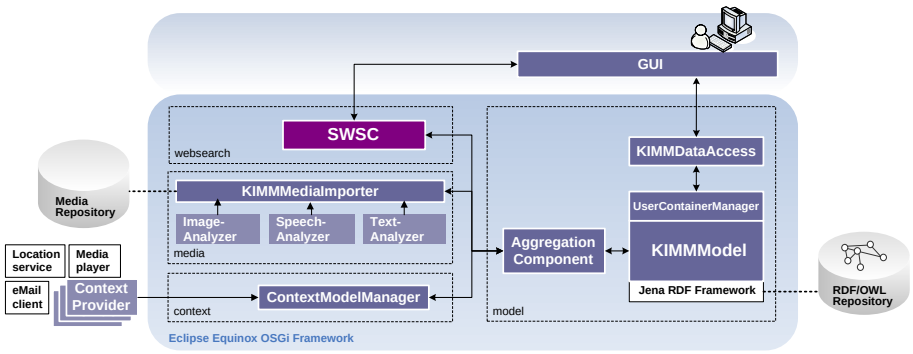


Fig. 3. The overall *K-IMM* System architecture.

All of the components are realized as plug-ins (bundles), according to the OSGi specification [19], and developed and run within the Eclipse Equinox execution framework. Thus, further plug-ins for specific media type analysis and processing or advanced components for visualization can easily be added to the system, and can be started and stopped dynamically at runtime. Hence, it is possible to run the system e.g. just for image management (starting only image analyzing and image semantics deducing components), or only with low-level

indexing (without semantic modeling), if desired. The media analyzing components extract available properties and features and pass them to the knowledge modeling components. In our prototype implementation RDF and OWL processing, storage and reasoning is based on the Jena Framework[7] including the Jena Inference Support. Further components, which are also not subject of this paper, comprise context aggregation and modeling.

4.1 Example: People Search

We set up a purpose-built graphical user interface (presented in Fig. 4) which shows the collection of documents (in this case images and text documents) managed by the underlying K-IMM System on the left, and extracted semantic entities (class instances based on our media management ontology) in the middle. In this example, the semantic entities were generated from people's names and locations, detected in the text documents using Named-Entity-Recognition methods and in metadata of the digital photographs. While the test set of text documents have been collected from the Internet and local desktop, the pictures were recent uploads at Flickr we downloaded via Flickr Web API. Thus, we could obtain Flickr users' names (i.e. first names, family names, and nicknames), EXIF metadata, and in some cases also location information from "geo-tagged" photos.

The semantic entities are represented in a categorized list. Clicking on the person entries starts the type specific Web search. Requests to the Web services were associated by default with the restriction to the appropriate file types (*FileTypeFilter*) and hosts (*HostFilter*) to limit the size of the initial result set. Additionally, we restricted the retrieved results from Swoogle even more, passing namespace filtering parameters (*ns:foaf*, etc.), which is of course not possible with Yahoo!. If exact match of the search phrase fails in a first try (initial query to the engine), the SWSC automatically retries the keywords with logical disjunction, to broaden the initial result set a bit, and leave further filtering to the document and term filters after download. We learned that this approach worked best in our case, although in most cases search results are "only" similar to the person we searched for (cf. Fig. 4: in this example we searched for "Mike Reinfeldt". After exact match failed, the SWSC broadens search to find similar names, in this case resulting in a set of other "Mikes").

The potential search results are represented in a list on the right side. Their representation is generated by the *ResultsFormatter* (mentioned in Section 3), which in this example returns a type-specific XHTML representation of the found people descriptions (mostly FOAF documents, their possibly contained *foaf:depiction* entries are used here to give a visual representation of the person). Clicking on other types of entities (e.g. a place) passes other types of search request to the SWSC, resulting in a specific *Query* object with according filter configuration (e.g. searching within address fields of vCard documents).

[7] `http://jena.sourceforge.net/`

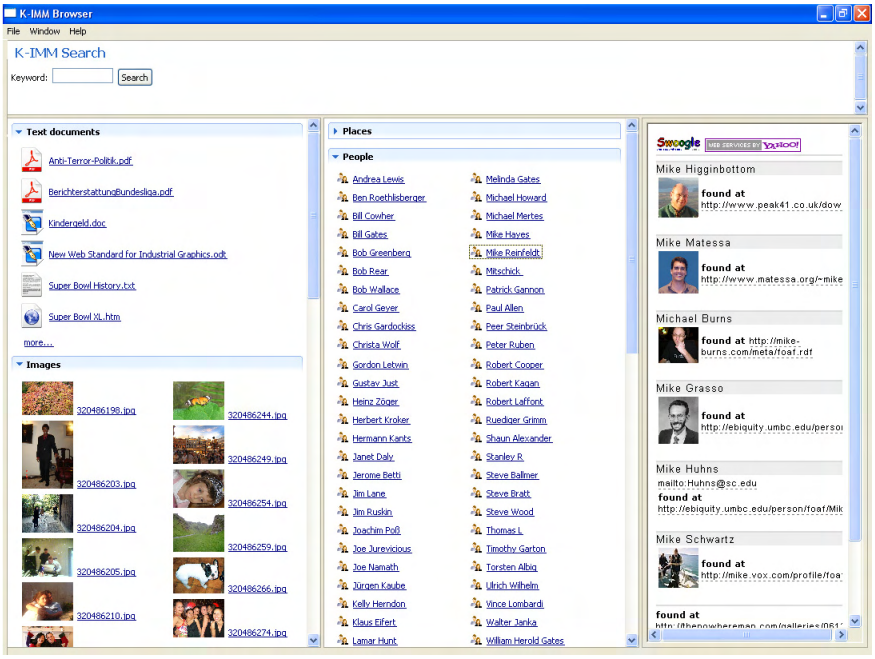


Fig. 4. Screenshots of the test search prototype, showing the results of a search example on the right.

4.2 Evaluation

Regarding the approximate usage of Semantic Web documents[8], *foaf* (`http://xmlns.com/foaf/0.1/`), *vcard* (`http://www.w3.org/2001/vcard-rdf/3.0#`), and *bio* (`http://purl.org/vocab/bio/0.1/`) are probably the most promising namespaces to find instance data describing people. Thus, people search was configured with an according *NamespaceFilter* white list and a collection of *TermNameFilter*s to evaluate whether found resources correspond to a person description ("name", "given name", "nick", "surname", etc.). To test and refine the settings of our query profile we ran a series of queries in batch mode based on a list of named entities (500 person names, i.e. first and last names), originally used for Named-Entity-Recognition. In doing so, we logged the number of initial, blocked, and accepted results, as well as the cause of the rejection, to get an idea of the quantity and quality of Semantic Web search with our implementation.

As to be seen in Fig. 5, aside from a few outliers, Swoogle returns - on average - a smaller initial result set, but with an overall higher value (less parse exceptions). Results from Yahoo! are more often blocked because of invalid content (non-RDF data). In general, document (namespace) and term filters restrict the result sets in both cases the most, as found search strings very often occur in

[8] `http://ebiquity.umbc.edu/blogger/100-most-common-rdf-namespaces/`

non-specific comments or labels not related to people descriptions. The analysis results are certainly quite evident, as Swoogle is much more dedicated to Semantic Web documents and uses a combination of Google meta-crawling, bounded HTML crawling, and RDF crawling [16]. On the other hand, we observed that in some cases Yahoo! retrieved documents which Swoogle did not find. Please note, that the application of less restrictive filters directly increases the number of accepted search results. That means, a quite high percentage of documents was actually usable Semantic Web content (available and valid RDF-based data), but blocked due to namespace or term filters for this special application scenario.

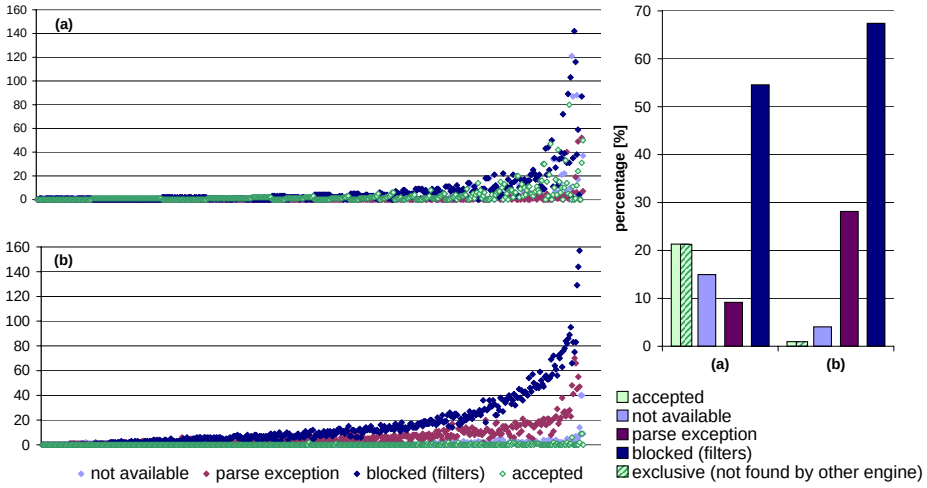


Fig. 5. An analysis of the results retrieved from Swoogle (a) and Yahoo! (b). On the left: absolute quantity (based on 500 people queries in March 2007, sorted along the x-axis with increasing number of initial results). On the right: relative distribution of the results quality. The *exclusive* results show, that the accepted result sets are almost disjoint.

However, there is of course a difference between tests (random lists of common named entities) and a real-world scenario of personal media annotation. People's social context is very individual, but in generally also more networked and interlinked (contacts and relationships). Thus, a general search within the range of the WWW would often fail (esp. regarding language difference). Instead, dedicated connections to community platforms (exploiting social networks), adjusted host filters, and specialized crawler implementations (e.g. using dictionaries for synonyms or different notations) should be used - which can be done within our SWSC.

In fact, today's WWW is still sparsely populated with Semantic Web content. Hence, search results are often not as expected. On the other hand, current

results are quite promising and show that user generated Semantic Web content (pushed by RDF-enabled community portals) is already retrievable and applicable. With a growing amount of Semantic Web content the developed SWSC can be configured to do more sophisticated filtering and ranking of obtained search results, e.g. using combined *TermName-* and *TermTypeFilters* to reduce false positives.

5 Conclusion and Future Work

In this paper we discussed Semantic Web search opportunities and their benefits within the context of the annotation of personal media collections. For that purpose we identified use cases of information search and integration, and developed a Semantic Web Search Component (SWSC) as a generic plug-in for various applications, using a combination of Web search engines for meta-crawling. As a particular usage scenario we presented a people search application with graphical user interface, based on our K-IMM media management system. Finally, we evaluated the search results of the implementation to show the particular benefits of this approach.

Our general approach allows further integration and extension of crawling implementations (e.g. to harvest community portals) for various scenarios and requirements with the help of customized query profiles and formatting rules. The current difficulty of our approach is basically the lack of valid and rich Semantic Web content indexed by available Web search engines. However, our component is capable of further application-specific refinements to use specialized or purpose-built Web services in combination, to extend the coverage of Semantic Web search. Furthermore, we think that Semantic Web content will increase in the next years with the help of communities and appropriate applications. Thus, our search component will support people and applications in discovering useful information resources in a growing Semantic Web.

Target of our future work will be the implementation of alternative connections and interfaces to other search engines or information sources to broaden the potential search results. As our evaluation shows, people search would certainly work much better with a dedicated FOAF search engine which collects FOAF data following *foaf:knows* links. Moreover, we are about to extend the developed component to realize different scenarios of information reuse, as we described in Section 2.2, e.g. search for public events, conferences, etc. We will also test proactive search scenarios and their benefits to users, which yet necessitates an acceleration of the evaluation and formatting of search results. Therefore, our current prototype already stores - in addition to the mentioned caching mechanism - lists of accepted documents which have been retrieved and evaluated beforehand in a background task.

References

1. J. J. Carroll and G. Klyne. Resource description framework (RDF): Concepts and abstract syntax. W3C recommendation, W3C, February 2004.

http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/.
2. D. McGuinness and F. van Harmelen. 2004 OWL web ontology language overview. W3c recommendation, W3C, 2004. http://www.w3.org/TR/owl-features/.
3. A. Chakravarthy, F. Ciravegna, and V. Lanfranchi. Cross-media document annotation and enrichment. In *Proceedings of the 1st Semantic Authoring and Annotation Workshop (SAAW2006)*, 2006. http://www.dcs.shef.ac.uk/ ajay/publications/paper-camera-workshop.pdf.
4. S. Handschuh and S. Staab. Authoring and annotation of web pages in cream. In *Proceedings of the Eleventh International World Wide Web Conference, WWW2002*, pages 462–473, 2002.
5. J. Kahan, M.R. Koivunen, E. Prud'hommeaux, and R.R. Swick. Annotea: an open rdf infrastructure for shared web annotations. In *Proceedings of the 10th International World Wide Web Conference*, pages 623–632, 2001.
6. S. Bloehdorn, K. Petridis, C. Saathoff, N. Simou, V. Tzouvaras, Y.S. Avrithis, S. Handschuh, I. Kompatsiaris, S. Staab, and M.G. Strintzis. Semantic annotation of images and videos for multimedia analysis. In *ESWC*, pages 592–607, 2005.
7. A. Mathes. Folksonomies - cooperative classification and communication through shared metadata, December 2004.
8. M. Biddulph. Crawling the semantic web. In *Proceedings of XML Europe 2004*, 2004.
9. E. Bozsak, M. Ehrig, S. Handschuh, A. Hotho, A. Maedche, B. Motik, D. Oberle, C. Schmitz, S. Staab, L. Stojanovic, N. Stojanovic, R. Studer, G. Stumme, Y. Sure, J. Tane, R. Volz, and V. Zacharias. Kaon - towards a large scale semantic web. In *EC-Web 2002, Aix-en-Provence, France, 2002, Proceedings*, volume 2455 of *LNCS*, pages 304–313. Springer, 2002.
10. L. Dodds. Slug: A semantic web crawler. In *Proceedings of Jena User Conference 2006*, 2006.
11. T. Finin, L. Ding, R. Pan, A. Joshi, P. Kolari, A. Java, and Y. Peng. Swoogle: Searching for knowledge on the Semantic Web. In *AAAI 05 (intelligent systems demo)*, July 2005.
12. R. T. Fielding. *Architectural styles and the design of network-based software architectures*. PhD thesis, University of California, Irvine, 2000.
13. A. Miles and D. Brickley. SKOS Core Guide. W3C Working Draft, World Wide Web Consortium, November 2005.
14. B. Adida and M. Birbeck. RDFa primer 1.0. W3C Editors' Draft, W3C, 2006. http://www.w3.org/2006/07/SWD/RDFa/primer/.
15. D. Brickley and R. V. Guha. RDF vocabulary description language 1.0: RDF schema. W3C recommendation, W3C, February 2004. http://www.w3.org/TR/2004/REC-rdf-schema-20040210/.
16. L. Ding and T. Finin. Characterizing the semantic web on the web. In *Proceedings of the 5th International Semantic Web Conference*, November 2006.
17. A. Mitschick. Ontology-based management of private multimedia collections: Meeting the demands of home users. In *6th International Conference on Knowledge Management (I-KNOW'06), Special Track on Advanced Semantic Technologies*, Graz, Austria, 9 2006.
18. A. Mitschick and K. Meiß ner. A stepwise modeling approach for individual media semantics. In *GI-Edition Lecture Notes in Informatics (LNI)*, Dresden, Germany, 10 2006.
19. D. Marples and P. Kriens. The open services gateway initiative: An introductory overview., 2001.

Combining Social and Semantic Metadata for Search in a Document Repository

Christoph Herzog[1], Michael Luger[2], and Marcus Herzog[3]

[1] E-Commerce Competence Center, EC3
christoph.herzog@ec3.at
[2] DERI Innsbruck
michael.luger@deri.org
[3] Vienna University of Technology
herzog@dbai.tuwien.ac.at

Abstract. With the success of social applications like Flickr, del.icio.us and YouTube, social software has become the focus of several research initiatives. Especially the idea of combining social and semantic aspects has been recently gaining significant attention in the semantic web community. In this paper we research and discuss the properties of metadata in social systems (folksonomies) compared to metadata in semantic systems (ontologies). We then present our idea for creating a link between a folksonomy and an ontology in order to combine the usability and flexibility of folksonomies with the precision of ontologies for a semantic search application. Our approach is motivated by the requirements of the OnTourism project, which has the goal of creatng a document repository which benefits from both ontology and folksonomy metadata.

1 Introduction

"Web 2.0", a term coined by Tim O'Reilly, has become a much debated topic in the web community. O'Reilly defines "Web 2.0" as platform of (web-based) software that incorporates user participation and "gets better the more people use it" [13]. With the remarkable success of social applications like YouTube, this idea has gained significant attention from the scientific community.

The strength of social applications is that they are easy to use and can generate massive amounts of metadata through an implicit community effort. However, these metadata are semantically not clearly defined and not suitable for reasoning or similar tasks. In this paper we explore the properties of social and semantic metadata. We also present our ideas for how to find relations between the two, based on their observable use in describing documents. Our work is motivated by the OnTourism project, which has the goal of creating a document repository that makes use of both semantic and social metadata.

The paper is organised as follows: In section 2 we review social and semantic metadata and give a comparison. In section 3 we present the OnTourism project and our ideas for a social semantic document repository. This is followed by a review of related work in section 4 and, finally, section 5 concludes the paper.

2 Comparing Social and Semantic Metadata

Folksonomies With the rising popularity of applications like *Flickr*[4], *del.icio.us*[5] and *YouTube*[6], social software has become a research topic. We think the two most important reasons for the high user acceptance of social software are:

1. *Low entry barriers* – Successful social tools like *Flickr* make it as easy as possible for the user to participate. Time, effort and cognitive cost required to use the system are minimised [9].
2. *Instant and delayed gratification* – The tools we examined exhibit patterns of what Ohmukai et al describe as *instant gratification* and *delayed gratification* [12]. Instant gratification is the direct and egoistic benefit users draw from using the system (i.e., organising their photos or bookmarks). Delayed gratification is the added value generated by the community.

Social software places an emphasis on users assigning freely chosen keywords to shared objects. While this is not a new idea, the novel aspect is that not only the author but, to a varying extent, also other users can assign keywords to an object. The tags (keywords) applied by users constitute an emergent vocabulary for which the term *Folksonomy* has been coined by Thomas Vander Wal [15].

A folksonomy in this sense is a set of terms, which from a mathematical point of view can be seen as a tripartite graph with hyper-edges, consisting of *(user, tag, object)* triples. The distribution of the tags follows a power law curve as Vander Wal points out in [16]. Figure 1 shows the tag distribution of a sample of approximately 200.000 tag usages from the del.icio.us folksonomy. The horizontal axis represents the approximately 1000 tags which are used at least 20 times. The vertical axis shows how often each tag is used.

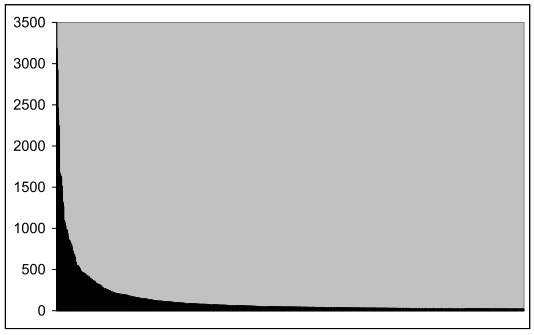


Fig. 1. Tag distribution of a sample from the del.icio.us folksonomy.

The direct benefit of folksonomies lies in making meaningful metadata available in an implicit community effort. This is especially true for systems like photo

[4] A web application for sharing photos. See http://www.flickr.com/
[5] A web-based social bookmarking tool. See http://del.icio.us/
[6] A web application for sharing videos. See http://www.youtube.com/

or video platforms, which are not amenable for text search methods. While the applied tags do not yield explicit semantics, they have the inherent benefit of "speaking the user's language", which is hard to accomplish by top-down ontology engineering. Furthermore, folksonomies conserve minority expressions. When regarding the power law distribution, the most common (*"strong"*) tags represent the major "desire lines" of the emergent vocabulary. However, the long tail of seldom used tags can contain highly specific terms, making the tagged objects amenable to being found by more unusual expressions.

Ontologies The semantic web initiative pursues the goal of creating data and metadata in such a way that not only humans but also machines can make use of it. The idea is that the *meaning* of the data should be expressed in a format which enables it to be processed by computers. Towards this goal, most systems make use of ontologies to describe their data or metadata. An ontology is a model of a real-world domain. This model specifies the most important concepts of that domain, their attributes and relations between concepts.

A particular usage of ontologies found in many semantic systems is the task of inferring new knowledge from facts and rules expressed in an ontology language. Another common task is the execution of search queries on data represented in an ontology language to retrieve semantically meaningful search results. The combination of both leads to semantic search applications that make full use of ontologies in order to provide complete and relevant answers to user queries.

Comparison Folksonomies and ontologies are targeted towards very different applications. In a direct comparison (see table 1), it becomes apparent that ontologies are more suitable for situations where a precise description of data is required and the cost of metadata creation is not an issue. Folksonomies on the other hand perform better when large quantities of metadata are required, where the metadata's precision is not of predominant importance.

3 OnTourism

Our idea of combining folksonomy and ontology metadata is motivated by our current work on the OnTourism project. The main goal of OnTourism is to implement a semantic search functionality on a call centre's existing document repository. The repository contains MS Word and PDF documents which are created by the call centre agents. The expected value of semantic search in this environment is that customer requests can be answered more quickly and more precisely. However, the intended users of the system are not experts in using ontologies, therefore, a social approach will be utilized in order to complement the semantic search function and make the ontology more accessible for the users.

Two kinds of metadata will be employed to desctibe the document's contents and target audience – free user selected keywords (tags) on the one hand and entities of a defined ontology on the other hand. In order to combine the advantages of the ontology and folksonomy metadata, statistical methods will be used

	Folksonomy	Ontology
Structure	flat	hierarchical structure
Creation	by users during the act of using the system	by ontology experts at a given time
Synonyms	no synonym control	synonym control possible
Precision	low precision	high precision
Flexibility	high flexibility	low flexibility
Creation Cost	low, created by users	high, created by experts
Change	highly dynamic, changes constantly	rigid, often has to be recreated to accommodate change
Usability	no expertise required	requires proficiency in handling
Vocabulary	users vocabulary	experts vocabulary
Scalability	works better in a large scale	works better in a small scale

Table 1. Comparison between folksonomies and ontologies (based on [1])

to find probable relations between folksonomy tags and ontology elements. Also, the search functionality itself will be a combination of the results of a semantic search utilising ontology reasoning and a search on the folksonomy tags.

3.1 User Driven Metadata – The OnTourism Folksonomy

Users of the system can add tags through a social bookmarking system. The incentive for adding tags is that the bookmarked documents can be found by searching for the assigned tags, which are the ones that for the user best describe the document. This contributes to the idea of "keeping found things found", i.e., being able to quickly re-find documents once discovered.

The main problem we face is the low number of users. In the call centre, approximately 15 people work with the document repository, limiting the user base. Through the annotation by the document's author, we make sure that each document is tagged at least by one person. However, in the call centre's set-up we must assume that many documents will be tagged by the author only and that even more popular documents receive tags from five or less users.

Another problem folksonomies as flat collections of terms face, is synonym control. People can use different tags with the same meaning (e.g., "Apple" vs. "Mac" vs. "Macintosh"). We do not seek a solution for synonym control in folksonomies, but rather intend to provide strong feedback to the user at the time of entering tags. When entering a keyword, the user will be given suggestions of likely matching or likely related folksonomy terms in real time. In this way we hope to achieve a quick convergence of the vocabulary.

3.2 Structured Metadata – The OnTourism Ontology

In addition to the approach of enriching the application with a social bookmarking functionality, a metadata ontology allows for the annotation of the docu-

ments on a more sophisticated and less ambiguous level. This ontology is aimed towards capturing the concepts and relations of the tourism domain as precisely and completely as required for our scenario. A semantic query engine enables to extract more accurate information than regular query engines that solely rely on information retrieval on a purely syntactical level or on unstructured tags.

The actual structure of the ontology is designed towards enabling the creation of an easy to use user interface, both for the process of annotation and for the semantic search component. Within the OnTourism project a close collaboration with Österreich Werbung[7] makes it possible to build the data model based on expert domain knowledge. Initially, the ontology broadly covers the tourism domain in low depth and the domains of special importance to the call centre application in more detail. A basic vocabulary for spatial-location related information such as the Basic Geo Vocabulary[8] is being considered to be incorporated into the ontology. One of the goals of this approach is to use the system's reasoning facilities in order to improve the output of location-related queries.

3.3 The Link between Social and Semantic

In the OnTourism system both ontology and folksonomy metadata will be incorporated. The ontology metadata provides the benefit of enabling a semantic search engine to find precise results and to apply reasoning procedures on the metadata. The folksonomy metadata provides the benefit of generating metadata in terms of a user-driven emergent vocabulary. Our goal, however, is to find relations between the folksonomy and the ontology metadata in such a way that in the overall system the strengths of both are emphasised.

We do not intend to combine the folksonomy and ontology metadata directly, but instead utilise the statistical relation between folksonomy tags and ontology elements, eventually using the folksonomy to enhance the usability of the ontology. Furthermore, semantic search results and the results of a search for tags will be merged into a combined search result.

From the analysis of the co-occurrence between folksonomy terms and ontology elements on single objects, we obtain a "statistical mapping" between the two types of metadata. The goal in extracting these relations is to find for any given tag from the folksonomy the most likely related entities from the ontology.

The user interface for annotating documents will utilise the folksonomy in order to help the user to find desired ontology elements. When entering keywords, the user is presented with suggestions for already existing tags. Once a user chooses such a tag, suggestions are presented for ontology elements most likely related to the keyword by the "mapping" discussed above. An ontology browser will enable the user to select ontology elements not suggsted yet.

Ontology elements which the system suggests may actually be only weakly related to the selected tag, but any useful suggestion improves the usability of the ontology. Moreover, if the suggested elements are not semantically related

[7] The Austrian national tourism organisation.
[8] Basic Geo (WGS84 lat/long) Vocabulary, http://www.w3.org/2003/01/geo/

then they may still be thematically related. In this way the suggestions may be useful for the user as a recommendation (e.g., a user who entered the tag "skiing" might also want to add the ontology element labelled "Mountain").

Mapping Folksonomy to Ontology The algorithm for relating ontology elements to folksonomy tags will be one of the main outputs of the OnTourism project. We shortly present some first ideas for this algorithm.

A standard relatedness measure like the jaccard coefficient [11] or cosine similarity [2] can be selected to define the relatedness between tags based on their co-occurence on documents. A similarity between ontology elements and folksonomy elements will be constructed equivalently.

However, considering the network of related tags from the folksonomy, connected by edges weighted with a relatedness measure as described above, we can extract a hierarchy of clusters and sub-clusters using network analysis methods. For example a specific approach towards this goal is described in [6]. For each cluster of tags in this hierarchy we compute the relatedness to a given ontology element as the sum of the relatedness to the individual tags in a given cluster.

Having done so, we can refine the list of related ontology elements for a given tag by going up along the hierarchy of clusters, adding the ontology elements related to the (bigger) parent cluster with a decreasing weight. In this way ontology elements directly related have a higher weight than ontology elements related by the increasingly fuzzy clusters. Through this procedure, more ontology elements are added to the list of candidates to be suggested to the user. This is especially useful if only few ontology elements co-occur with a given tag.

Those suggested ontology elements actually selected by the user are very likely to have a strong relation to the tag entered by the user. Therefore, the user's choice should strengthen the statistical relation between the tag and the selected ontology element. We are currently investigating how this is best achieved.

3.4 Social Semantic Search

In the OnTourism system, searching for documents will be a combination of semantic search, search on folksonomy terms and full text search. The three search methods can be executed in parallel, with the complete search result being a weighted combination of the three separate (possibly empty) result sets.

Semantic Search — As described above, the semantic search application will provide a graphical interface that allows the user to select concrete objects and attributes from the underlying semantic data model. The actual parameters entered through this interface are then translated into the corresponding formal query, which is then performed on top of the ontology storage component. This semantic search is expected to yield results of a precision not achievable by performing the query upon potentially ambiguous tags.

Folksonomy Search — The second component is the search for documents annotated with specific tags. Where semantic search may fail to retrieve some relevant documents by being too restrictive, searching for folksonomy terms can

provide more generous results while still being relevant for the annotated document. Search and ranking in the folksonomy metadata will be based on the *FolkRank* algorithm introduced in [5]. The results of the search on folksonomy metadata will have a lower weight than those of the semantic search.

Full Text Search — The results of full text search will also be considered in order to make sparsely annotated documents retrievable. The weight of the full text search will be considerably smaller than that of the other two methods.

4 Related Work

Several works are being investigated towards the goal of achieving a synergy between social and semantic applications. These works mainly follow one of two approaches [14]: adding more precise semantics to social systems or using a community of users to enhance semantic software.

Examples for adding more semantics to social systems include the idea of semantic enrichment of tags in weblogs [4] or, more generally, the idea to extend the *(object, tag)* graph of a folksonomy towards an *(object, ontology node, tag)* graph [7]. Examples for attempts to add some of the benefit of folksonomies to ontologies include ideas to extend ontologies in a folksonomy-like approach [3] or to add multiple labels to ontology nodes, an idea formulated by Maedche [8].

Another line of works is concerned with extracting semantic relations from folksonomies. While the extraction of complete ontologies from folksonomies appears to be a rather less explored area, there are several works towards extracting at least basic taxonomies from folksonomies [6, 10, 17].

5 Conclusion

Social and semantic software each have their own strengths and weaknesses. Social software is based on a low effort for the individual user in participating. Such systems can generate massive amounts of meaningful metadata. These metadata, however, are neither structured nor controlled. Ontology-based metadata on the other hand has a clear semantic meaning. Creating such semantically rich metadata, however, is expensive in terms of the annotation effort.

In this paper we presented our idea of social semantic document repository, motivated by the requirements of the OnTourism project. In this repository, documents will be annotated with both user defined keywords from an emergent vocabulary (folksonomy) and with metadata from an ontology's vocabulary.

We intend to find relations between the tags and the ontology elements, identified through correlations of the two kinds of metadata when used to annotate single documents. We will use these relations in order to make the ontology more accessible for the users through an appropriate user interface.

Furthermore, in order to search for documents, we will combine the results from search on folksonomy metadata and from a semantic search engine. In this way the search application benefits from both the precision of the ontology and the flexibility of the folksonomy generated by the social component.

References

1. Christine Albrecht. Folksonomy. Master's thesis, Vienna University of Technology, Vienna, Austria, March 2006.
2. Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, June 1999.
3. Scott Bateman, Christopher Brooks, and Gord McCalla. Collaborative tagging approaches for ontological metadata in adaptive e-learning systems. In *Proceedings of the 4th International Workshop on Applications of Semantic Web Technologies for E-Learning (SWEL'06)*, June 2006.
4. Steve Cayzer. What next for semantic blogging? In *Proceedings of the SEMANTICS 2006 conference*, pages 71–81, Vienna, Austria, November 2006.
5. Andreas Hotho, Robert Jschke, Christoph Schmitz, and Gerd Stumme. Information retrieval in folksonomies: Search and ranking. In *Proceedings of the 3rd European Semantic Web Conference (ESWC2006)*, pages 411–426, Budva, Montenegro, 2006.
6. Renaud Lambiotte and Marcel Ausloos. Collaborative tagging as a tripartite network. *ArXiv Computer Science e-prints*, December 2005.
7. K. Faith Lawrence and M. C. Schraefel. Freedom and restraint: Tags, vocabularies and ontologies. In *Proceedings of the 2nd IEEE International Conference on Information & Communication Technologies*, Damascus, Syria, 2006.
8. Alexander Maedche. Emergent semantics for ontologies – support by an explicit lexical layer and ontology learning. *IEEE Intelligent Systems - Trends & Controversies*, pages 78–86, February 2002.
9. Adam Mathes. Folksonomies – cooperative classification and communication through shared metadata. http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html, December 2004.
10. Peter Mika. Ontologies are us: A unified model of social networks and semantics. In *Proceedings of the International Semantic Web Conference 2005 (ISWC'05)*, pages 522–536, Galway, Ireland, November 2005.
11. Junichiro Mori, Yutaka Matsuo, Mitsuru Ishizuka, and Boi Faltings. Keyword extraction from the web for person metadata annotation. In *Proceedings of the 4th International Workshop on Knowledge Markup and Semantic Annotation, (ISWC2004)*, pages 51–60, Hiroshima, Japan, November 2004.
12. Ikki Ohmukai, Masahiro Hamasaki, and Hideaki Takeda. A proposal of community-based folksonomy with rdf metadata. In *Proceedings of the Workshop on End User Semantic Web Interaction, (ISWC2005)*, Galeway, Ireland, November 2005.
13. Tim O'Reilly. Web 2.0: Compact definition? http://radar.oreilly.com/archives/-2005/10/web_20_compact_definition.html, October 2005.
14. Sebastian Schaffert. Semantic social software: Semantically enabled social software or socially enabled semantic web? In *Proceedings of the SEMANTICS 2006 conference*, pages 99–112, Vienna, Austria, November 2006. OCG.
15. Gene Smith. Folksonomy: Social classification. http://atomiq.org/archives/2004/-08/folksonomy_social_classification.html, August 2004.
16. Thomas Vander Wal. Explaining and showing broad and narrow folksonomies. http://www.personalinfocloud.com/2005/02/explaining_and_.html, February 2005.
17. Xian Wu, Lei Zhang, and Yong Yu. Exploring social annotations for the semantic web. In *Proceedings of the 15th International Conference on World Wide Web (WWW'06)*, pages 417–426, Edinburgh, Scotland, May 2006. ACM Press.

Recommending Smart Tags in a Social Bookmarking System

Pierpaolo Basile, Domenico Gendarmi, Filippo Lanubile, Giovanni Semeraro

University of Bari,
Dipartimento di Informatica,
Via Orabona, 4, 70126 - Bari, Italy
{basilepp,gendarmi,lanubile,semeraro}@di.uniba.it

Abstract. Collaborative tagging systems are harnessing the power of online communities, making the task of knowledge contribution more attractive to a broader audience of Web users. In particular, social bookmarking systems have shifted the organization of bookmarks from an individual activity performed on a personal desktop to a collective endeavor over the Web. In such a context, suggestive tagging has proved to be helpful in consolidating the usage of tags, leading to a quick convergence to a folksonomy.
In a social bookmarking system, users' annotations can be regarded as a reliable indicator of interests and preferences. A recommender system is able to learn user interests and preferences during the interaction in order to construct a user profile.
In this paper, we propose a smart tag recommender able to learn from past user interaction as well as the content of the resources to annotate. The aim of the system is to support users of current social bookmarking systems by providing a list of new meaningful tags. The proposed system is based on ITem Recommender, a content-based recommender previously used in a Digital Library scenario.

Keywords: collaborative tagging, folksonomy, recommender system, semantic web, user profile, suggestive tagging, social bookmaking

1 Introduction

Since Tim Berners-Lee's inceptive Semantic Web vision [2], online communities have taken an active role in the task of knowledge contribution on the Web. Users are no longer passive information consumers, but active participants working in close collaboration to create new content and share it, using the Web as the underlying platform. The phenomenon of Web 2.0[1] has led to the development of several tools which have succeeded in making this task more attractive to a broader audience.

[1] Tim O'Reilly: What is Web 2.0. Design Patterns and Business Models for the Next Generation of Software. http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html, 2005.

Powerful tools for lightweight metadata creation, such as collaborative tagging systems, harness the power of virtual communities and have been shown effective in gathering quickly large amounts of information directly generated by users.

Collaborative tagging systems, also known as folksonomies [8], allow people to organize a set of resources, annotating them with tags via a web-based interface. Unlike top-down centralized approaches, folksonomies have revealed a noteworthy ability in adhering to the personal way of thinking [7]. The opportunity of using free tags with no restrictions allows users to express their own perspective on the annotated resource. Therefore, these annotations can become a reliable indicator of interests and preferences of active participants in such systems.

On the other hand, recommender systems [5] are able to learn user interests during the interaction in order to construct (and update) a user profile that can be later exploited for information filtering. A recommender system can be improved by the sheer size of the content available on the Web and the diverse expectations of its user base. Web applications need to combine all available knowledge in order to provide personalized and user-friendly services. Over the years, personalized Web applications and services have been developed, which exploit Web Mining technologies to discover shallow patterns hidden within masses of transactional, navigational, and content-structural data. In addition, knowledge-based recommender systems are able to exploit domain knowledge by integrating domain ontologies.

We think that combining the strengths of Web Mining with the benefit of deeper semantic and the attractiveness of collaborative tagging systems can be a first step to bridge the gap between Semantic Web and Web 2.0.

In this paper we propose an approach to improve an existing recommender system with the purpose of exploiting the information about users' interests provided in form of tags by del.icio.us[2], the most popular social bookmarking system. Our aim is to support users of current collaborative tagging systems by providing tag recommendations based on both the annotations already performed and the content to annotate. The contribution is twofold: A semantic suggesting feature in a social bookmarking system can foster the tag convergence, useful for example to limit the synonymy issue; furthermore, suggesting meaningful tags to a user according to the interests stored in her profile can significantly improve the user experience, augmenting the number of active participants in the collaborative system.

The remainder of the paper is structured as follows. Section 2 presents background information about tag recommendation in social bookmarking systems. An illustrative user scenario motivating our approach is provided in Section 3, while Section 4 describes how we plan to extend our recommender system. Finally, Section 5 draws conclusions and points out some challenges we are going to address in the near future.

2 Related Work

Previous studies on bookmarks use showed that main motivations for creating bookmarks are based on personal interests and quality of the content, high frequency

[2] http://del.icio.us

of current use, as well as a sense of potential reuse [1]. The most familiar approach to store markers for re-finding information on the Web has been through the use of personal bookmarks, supported by almost all browsers. In the last few years, social bookmarking systems have shifted the organization of bookmarks from an individual activity performed on a personal desktop to a collective endeavor over the Web.

Although bookmark collections are personal, the opportunity of accessing to such personal collections from any Web-connected machine (together with the use of free multiple tags, helpful in overcoming the limitation of the traditional hierarchically organized folders) have led to a wide spread of these social systems. Even though contributions are motivated by the private need to easily organize personal items, they also aggregate at a higher level via a collaborative tagging endeavor, that allows the shaping of social networks [13]. Furthermore, some tagging support features, such as suggestive tagging [11], have proved to be helpful in improving the user experience as well as fostering an emerging consensus on the meaning of the terms rising up in the folksonomy [6].

Among the different social bookmarking systems, del.icio.us, one of the earliest and most popular ones, is the only application that illustrates some remarkable suggestive tagging features. When a user saves a bookmark in del.icio.us, she can manually enter as many tags as she would like, but she can also be supported by a list of suggested tags (Figure 1). Popular tags are what other people have tagged this page as, and recommended tags are a combination of tags user has already used and tags that other people have used.

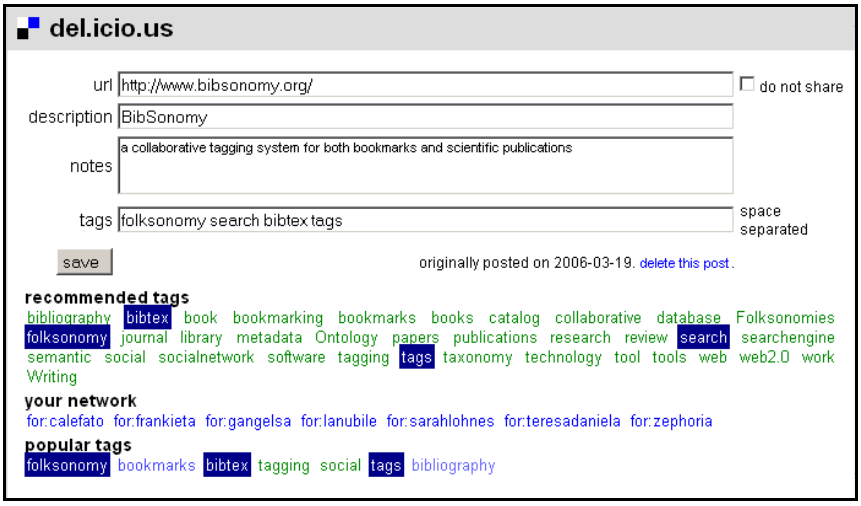


Figure 1. Saving a bookmark in del.icio.us

Rather than recommendations based on some underlying analysis, this kind of suggestions can be regarded as a selection of tags in the sense that the system has to choose a small number of tags to display among the sheer size of terms already associated to an item. According to the tag selection approach, Sen et al. [16]

investigate how different algorithms for selecting tags to display, influence users' personal vocabularies while annotating movies in a movie recommendation system.

On the other hand, as an evidence of the lack of social bookmarking systems that exploit actual tag recommendations (as far as we know), there is few work on such a topic published via the scholarly literature.

Xu et al. [17] define a set of general criteria for a good tagging system to identify the most appropriate tags, while eliminating noise and spam. These criteria, identified through a study of tag usage by real users in My Web 2.0, cover desirable properties of a good tagging system, including high coverage of multiple facets to ensure good recall, least effort to reduce the cost involved in browsing, and high popularity to ensure tag quality. The authors then propose a collaborative tag suggestion algorithm that adopts those criteria to recommend appropriate tags.

Hotho et al. [9] propose an adaptation of both a data mining and information retrieval approach to detect emergent semantics within a collaborative tagging system. The first adaptation lies in reducing the three-dimensional folksonomy to a two-dimensional formal context in order to apply association rule mining techniques. Discovered association rules can be then exploited in a recommender system which supports the user in choosing useful tags. The latter is an adaptation of the PageRank algorithm [3] to the tripartite hypergraph structure of a folksonomy. The algorithm, named FolkRank, incorporates the idea that a node is important if there are many edges from other nodes pointing to it and if those nodes are important themselves, and applies the same principle to the tripartite graph of the folksonomy. The FolkRank algorithm is then used to rank users, tags, resources by their importance. Authors suggest that such rankings can be exploited to generate recommendations for each user about new potential resources of interest, related tags and other users possibly interested on analogous topics.

3 Motivating Scenario

We consider del.icio.us as reference system because of the huge number of registered users and the richness of suggestive tagging. In our scenario, John is a novice user, who has just registered into the system and has no stored bookmarks yet. When John is going to save his first bookmark, the current system suggests popular tags, i.e., terms heavily used by other users to annotate the same resource. A recommender system cannot suggest anything, until the user provides enough information to generate a profile delineating personal interests. However, the use of del.icio.us as an underlying platform makes it possible to support John with popular suggested tags, until the recommender becomes able to actually learn John's interests on the strength of his personal bookmarks and tags.

After John has been using del.icio.us for a while, he has progressively built a large bookmark collection, as well as a rich vocabulary of personal tags that can be exploited by the Smart Tag Recommender system.

When John wishes to save a new bookmark in his personal space, he has a chance to reuse some tags previously used, but he might also enter new tags according to the subject of the resource he is going to annotate. This time the Smart Tag

Recommender can analyze the content of the resource selected by John in order to obtain a collection of concepts describing the bookmark. The output of the content analysis can be then used to retrieve similar bookmarks already annotated by John and find out which tags John has previously used to store such references.

According to the concepts extracted by the analyzer and the tags associated to existing similar bookmarks in John's user profile, the Smart Tag Recommender can now suggest meaningful tags for the resource John wishes to store. The Smart Tag Recommender is not intended to replace the existing del.icio.us recommender, since it provides a new layer of recommendation based on personal profiles and not on popularity.

4 Recommender Architecture

The proposed scenario can be supported by a service that relies on a content-based recommender system, such as ITem Recommender (ITR) [10]. Indeed, this system is able to induce a profile of the user by learning from the content of documents she annotated with a feedback according to her preferences. The induced user profile is a structured representation of user interests which is then exploited to decide whether a new document fits in with the user's preferences. In our case, we consider the problem of learning user profiles as a binary text categorization task [14]: Each document has to be classified as interesting or not with respect to the user preferences. Therefore, the set of categories is $C = \{c_+, c_-\}$, where c_+ is the positive class (*user-likes*) and c_- the negative one (*user-dislikes*). ITR uses a Naïve Bayes method to text categorization; in this way the learned probabilistic model is used to classify a document d_i by selecting the class with the highest probability. As a working model for the Naïve Bayes classifier, we use the multinomial event model [12] to estimate the *a posteriori* probability, $P(c_j|d_i)$ of document d_i belonging to class c_j.

In order to capture the semantics of the user interests, learning is performed on documents that have been previously analyzed by advanced Natural Language Processing (NLP) techniques (implemented in the Content Analyzer module in Figure 2) able to discover relevant concepts representing the content of the documents. The key step in this process is Word Sense Disambiguation (WSD), which is the task of assigning a word with the most appropriate meaning, by taking into account the context (a set of words that precede and follow the word to be disambiguated) in which the word appears. To sum up, documents are represented by concepts instead of keywords, as in the classical vector space model [4]. In order to recognize correctly the meaning of the words, the WSD procedure relies on the WordNet lexical database, in which the set of all possible meanings for each word is maintained. Moreover, the WSD procedure will be integrated with an Entity Recognizer module in order to identify Named Entities that do not occur in WordNet. More details on the ITR system and the WSD procedure are reported in [15].

The ITR system can be easily adapted to the scenario of del.icio.us tags recommending. Indeed, ITR can be used to build a user profile able to support the user in the task of annotating resources by suggesting tags on the basis of previously tagged documents. Given $T = \{t_1, t_2, \ldots, t_n\}$, the set of all tags employed by the user in

her "tagging history", the idea is to include a set of n binary classifiers, each classifier c_k corresponding to tag t_k, in the user profile. Any new document d is then matched against the user profile so that each classifier c_k in the profile can predict whether d should be annotated with t_k. The final outcome of the matching process is the set of tags recommended by the classifiers in the user profile.

The set of documents used to train ITR is the set of all the documents previously annotated by the user. Each training document tagged with t_k is considered as a positive example for c_k, while the set of negative examples for c_k is represented by all documents that have not been tagged with t_k.

Figure 2 shows the conceptual architecture of the Smart Tag Recommender system. Full rows indicate the learning step, while dotted rows indicate the classification step.:

a) *Learning step:* An annotated documents is processed by the *Content Analyzer* in order to obtain the Bag-Of-Synsets (BOS) model of the document. To this purpose, NLP techniques, including WSD, are exploited. After that, for each tag t_k the *Profile Extractor* builds the corresponding classifier c_k, that will be part of the *User Profile*.

b) *Classification step:* A new document *(New Doc*) is processed by the *Content Analyzer*, then the *Recommender* uses *User Profile* to select the most appropriated tags for the document. Specifically, for each tag t_k *New Doc* is classified using the corresponding classifiers c_k. The output of this process is the list of recommended tags.

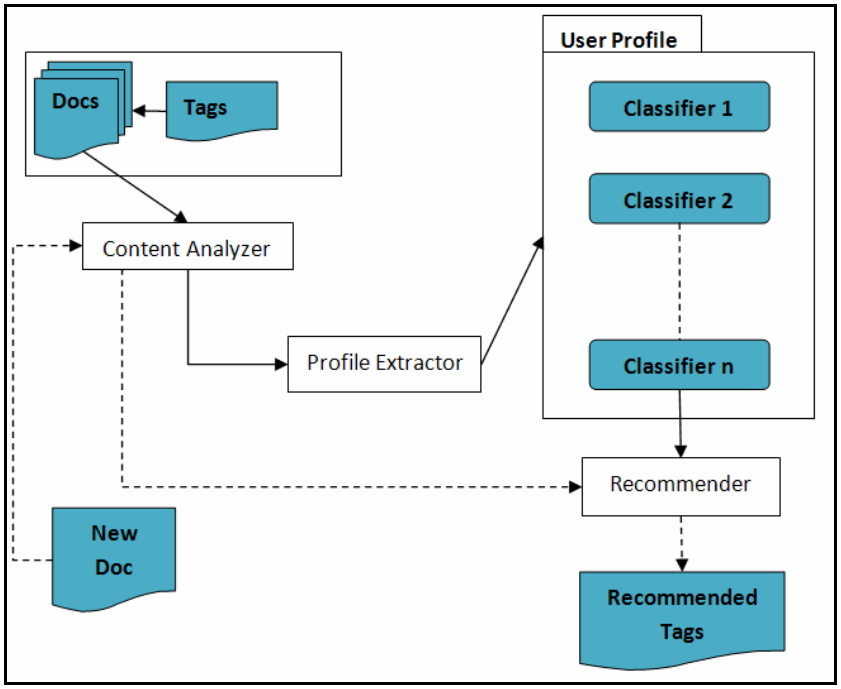


Figure 2. Smart Tag Recommender architecture

5 Conclusions and Future Work

Web 2.0 applications provide chance to semantically exploit the sheer size of user-generated content. Tags in a social bookmarking system can reveal users' interests and preferences. However, current systems suggest a lot of irrelevant tags, either on the basis of personal recent use or because of their popularity among the community. Our aim is to combine the strengths of Semantic Web and Web 2.0 in order to provide better personalized tag recommendations.

In this paper, we have described a strategy to design an intelligent recommender system which is able to learn from both past user interaction and the content of the resources to annotate. The system is based on an existing content-based recommender, that has been previously used in a Digital Library scenario. The main idea is presented in the context of del.icio.us, the most popular social bookmarking system. As future work, we plan to complete the development of the new recommender system and perform an experimental evaluation within del.icio.us, having the basic suggested tagging feature as a control group.

References

1. Abrams, D., Baecker, R., Chignell, M.: Information archiving with bookmarks: personal web space construction and organization. Proceedings of the SIGCHI conference on Human factors in computing systems, (1998), 41–48
2. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American (2001).
3. Brin, S., Page, L. The anatomy of a large-scale hypertextual Web search engine. Computer Networks and ISDN Systems, 30 (1–7) (1998), 107–117
4. Bruce, K.B., Cardelli, L., Pierce, B.C.: Comparing Object Encodings. In: Abadi, M., Ito, T. (eds.): Theoretical Aspects of Computer Software. Lecture Notes in Computer Science, Vol. 1281. Springer-Verlag, Berlin Heidelberg New York (1997), 415–438
5. Burke, R.: Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction, 12 (4) (2002), 331–370
6. Gendarmi D., Abbattista F., Lanubile F.: Fostering knowledge evolution through community-based participation. Proceedings of the 1st Workshop on Social and Collaborative Construction of Structured Knowledge at WWW'07, (2007)
7. Gendarmi D., Lanubile F.: Community-driven ontology evolution based on folksonomies. OTM Workshops, LNCS, Vol. 4277. Springer-Verlag, (2006), 181–188
8. Golder, S., Huberman, B.: Usage patterns of collaborative tagging systems. Journal of Information Science, 32(2) (2006), 198-208
9. Hotho, A., Jäschke, R., Schmitz, C., Stumme, G.: Emergent semantics in Bibsonomy. Proceedings of Workshop on Applications of Semantic Technologies, Informatik 2006. Lecture Notes in Informatics, (2006)
10. Lops, P., Degemmis, M., Semeraro, G.: Improving social filtering techniques through Wordnet-based user profiles. Proceedings of 11th International Conference on User Modeling, (2007)
11. Marlow, C., Naaman, M., Boyd, D., Davis, M.: HT06, tagging paper, taxonomy, Flickr, academic article, to read. Proceedings of the Seventeenth Conference on Hypertext and Hypermedia. (2006), 31–40

12. McCallum, A., Nigam, K.: A comparison of event models for naive bayes text classification. Proceedings of the AAAI/ICML-98 Workshop on Learning for Text Categorization, (1998), 41–48
13. Mika, P.: Ontologies are us: A unified model of social networks and semantics. Proceedings of the 4th International Semantic Web Conference, LNCS, Vol. 3729. Springer-Verlag, (2005) 522-536
14. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys, 34(1), (2002)
15. Semeraro, G., Degemmis, M., Lops, P., Basile, P.: Combining learning and word sense disambiguation for intelligent user profiling. Proceedings of twentieth International Joint Conference on Artificial Intelligence, (2007), 2856–2861
16. Sen, S., Lam, S. K., Rashid, A., Cosley, D., Frankowski, D., Osterhouse, J., Harper, F. M., Riedl, J.: Tagging, communities, vocabulary, evolution. Proceedings of the 20th Anniversary Conference on Computer Supported Cooperative Work, (2006), 181-190
17. Xu, Z., Fu, Y., Mao, J., Su, D.: Towards the Semantic Web: Collaborative Tag Suggestions. Proceedings of Collaborative Web Tagging Workshop at 15th International World Wide Web Conference (2006).

Bridging the Gap Between Folksonomies and the Semantic Web: An Experience Report

Sofia Angeletou[1], Marta Sabou[1], Lucia Specia[2], and Enrico Motta[1]

[1] Knowledge Media Institute (KMi)
The Open University, Milton Keynes, United Kingdom
{S.Angeletou, R.M.Sabou, E.Motta}@open.ac.uk
[2] Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo, São Carlos, Brazil
lspecia@icmc.usp.br

Abstract. While folksonomies allow tagging of similar resources with a variety of tags, their content retrieval mechanisms are severely hampered by being agnostic to the relations that exist between these tags. To overcome this limitation, several methods have been proposed to find groups of *implicitly* inter-related tags. We believe that content retrieval can be further improved by making the relations between tags *explicit*. In this paper we propose the semantic enrichment of folksonomy tags with explicit relations by *harvesting the Semantic Web*, i.e., dynamically selecting and combining relevant bits of knowledge from online ontologies. Our experimental results show that, while semantic enrichment needs to be aware of the particular characteristics of folksonomies and the Semantic Web, it is beneficial for both.

1 Introduction

Folksonomies [13] are typical Web2.0 systems that allow users to upload, tag and share content such as pictures, bookmarks etc. One of their distinctive features is that they are open, uncontrolled systems where users can annotate resources with different tags depending on their social or cultural backgrounds, expertise and perception of the world [2, 3, 9, 14]. For example, a zoologist can tag a photograph of a lion with {`felidae, pantherinae, mammal`}, while a non-zoology expert can use {`lion, king, animal, jungle`} for the same purpose. This freedom of tagging largely contributed to the success of folksonomies: users need neither to have prior knowledge or specific skills to use the system [5, 15], nor need to rely on a priori agreed structure or shared vocabulary.

Unfortunately, the simplistic tag-based search used by folksonomies is agnostic to the way tags relate to each other although they annotate the same or similar resources. For example, a search for {`mammal`} ignores all resources that have not been tagged with this specific word, even if they are tagged with related concepts such as {`lion, cow, cat`}. As a result, content retrieval activities such as searching, subscription and exploration are limited [2], they provide low-recall and hardly lend themselves to query-refinement [11]. Therefore, to

obtain satisfactory results, a searcher needs to build multiple complex queries to cover all the possible tags that could have been used by taggers [3, 9, 14]. As searchers rely on their own view about what inter-related tags best describe the resource they are looking for, it follows that content retrieval could be enhanced if folksonomies were aware of the relations between their tags.

Following this intuition, a variety of approaches have been proposed to identify inter-related tags. The existing work considers tag co-occurrence for the organisation of related tags into clusters. For example, [11] uses a subsumption-based model, derived from the co-occurrence of tags, to find groups or related tags. [2] organises the tag space as an undirected graph, representing co-occurring tags as vertices, weighting the edges between them according to their co-occurrence frequency, and applying a spectral clustering algorithm to refine the resulting groups. [15] uses a probabilistic model to generate groups of semantically related tags based on the co-occurrence of tags, resources, and users. These are represented as a multi-dimensional vector, where each dimension refers to a category of knowledge. Both the number of dimensions and the relation values of entities to each dimension are determined using log-likelihood estimates. [7] uses co-occurrence information to build graphs relating tags with users and tags with resources, and applies techniques of network analysis to discover sets of clusters of semantically related tags. [12] groups tags according to their co-occurrence using a clustering algorithm similar to clustering by committee [8]. Finally, most of the folksonomies provide funtionalities to derive "clusters" and "related tags", which apparently also rely on co-occurrence information and clustering techniques.

All the approaches, except from [12], focus on finding groups of related tags rather than identifying the semantics of those relations. In this work the authors envisaged tag space enrichment with semantic relations by exploring online ontologies. Their preliminary experiments on Flickr and Del.icio.us data confirmed that this is a promising strategy. Indeed, the recent growth of the Semantic Web has resulted in an increased amount of online available semantic data and has led to the first search engine to exploit this data, Swoogle [6]. These facts made it possible to build applications that *harvest the Semantic Web* (i.e., dynamically select, combine and exploit online knowledge) to successfully solve a variety of tasks, such as query disambiguation [4] and ontology matching [10].

Applying this novel paradigm to folksonomies would make them explicitly aware of the inherent semantic relations between their tags. For example, subsumption relations such as the ones depicted in Fig. 1 could be derived between the tags of the cluster {`lion, animal, mammal, feline, tiger`} by combining information from different online ontologies. The knowledge that *Lions* and *Tigers* are *kind of Mammals* would expand the potential of folksonomies. Users could make generic queries such as *"Return all mammals"* and obtain all the resources tagged with `lion` or `tiger` even if they are not explicitly tagged with `mammal` .

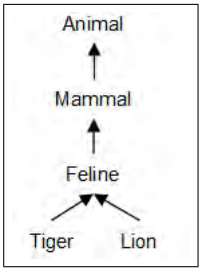


Fig. 1: Related Tags.

While previous work has experimentally shown that harvesting online knowledge yields good results when applied to ontologies [10], the folksonomy tag enrichment algorithm proposed in [12] was not fully automated. Therefore, an important research question is: *Can we enrich folksonomies by automatically harvesting the Semantic Web?* In particular, we are interested in finding out: *What are the major characteristics of the Semantic Web and folksonomies that need to be taken into account to perform such enrichment?* And if this enrichment is possible: *What are its benefits?* To answer these questions, we propose a method to enrich the tag space of folksonomies which assumes the existence of previously defined groups of potentially related tags (these can be obtained by any of the above mentioned techniques) and which is entirely focused on the exploitation of the Semantic Web (Section 2). This approach is automated by using the algorithm described in [10]. We present and discuss our experimental results which give an insight in the major characteristics of the Semantic Web and folksonomies that need to be considered when performing such enrichment (Section 3). We conclude and point out future work in Section 4.

2 Semantic Enrichment of Folksonomy Tag Space

In this section we describe our approach for semantically enriching the folksonomic tag spaces. Our method is based on [12], which describes a hybrid approach that combines harvesting the Semantic Web with using other Web resources such as Wikipedia and Google. As the goal of our work is to understand the potential and limitations of the Semantic Web when used to semantically enrich folksonomies, we have modified their algorithm so that it only relies on online ontologies. Our algorithm, presented next, takes as input a cluster of implicitly related tags and returns 1) a knowledge structure obtained by making explicit the semantic relations among them and 2) a set of tags which could not be semantically related to any other tag in their cluster or were not covered by the Semantic Web.

2.1 Semantic Enrichment Method

The semantic enrichment of each cluster is depicted in Fig. 2 and consists of two phases: Phase 1, concept definition for each tag (i.e., linking tags to ontology concepts) and Phase 2, relation discovery between all the possible pairs of tags.

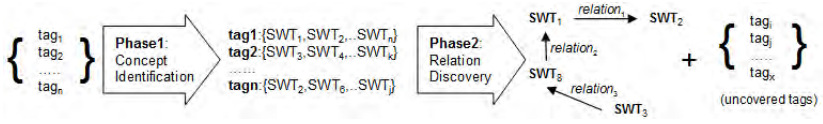


Fig. 2. Semantic Enrichment Method

Phase 1. Concept Identification: The first step *explicitly defines the meaning* of each tag by extracting all Semantic Web Terms (SWT) whose label or localname matches the tag. The matching between the tag and the SWT can be achieved using anchoring techniques ranging from strict to flexible string matching as described in [10].

Using the Semantic Web for extracting concepts is proposed in the work of [4] as a first step to query disambiguation. The authors search for candidate senses in online ontologies and then perform disambiguation based on the semantic similarity of the retrieved senses (e.g., `bass` can refer to either a fish or musical notes depending on the context in which it is used). While we use the same technique for SWT identification we do not explicitly disambiguate between them. In our case, disambiguation is a side effect of relation discovery (Phase 2).

The disambiguation of the tag sense (i.e., finding the right concept for a tag given its context) is approached differently in [12]. The authors rely on the heuristic that if pairs of tags from a cluster appear in the same ontology, then this leads to an implicit disambiguation (i.e., searching for `apple` and `fruit` leads to ontologies about fruits, while when searching for `apple` and `computer` they identify ontologies about computers). While this intuition holds in the case of domain-specific ontologies, it is problematic when the tags appear in broad, cross-domain ontologies such as WordNet[3] or TAP[4]. Also, by considering only ontologies that contain both tags, this approach potentially misses important information that might be declared in ontologies defining only one of the tags. This information can prove to be useful when combined with information from other ontologies. For example, an ontology containing *Apple* and *Mac*, can be combined with information from another ontology containing information about *Mac* and *Computer*. For these reasons, we retrieve all the potential SWTs for each tag and discover relations between them in Phase 2.

Phase 2. Relation Discovery: This step identifies *explicit semantic relations* among all the pairs of SWTs (`T1` and `T2`) discovered in the previous phase:

- **Subsumption Relations:** when one of the two SWTs is a subclass of the other, `T1 subClassOf T2`. This relation can be either declared in an ontology or derived by different levels of inference (no inference, basic transitivity, Description Logics reasoning). An example of inferred relation is: if `T1 subClassOf T2` and `T2 subClassOf T3` then `T1 subClassOf T3`.
- **Disjointness Relations:** when `T1` and `T2` are disjoint, `T1 disjointWith T2`. Again this relation can be declared or inferred. We use the algorithm described in Section 2.2 to discover disjointness and subsumption relations.
- **Generic Relations:** when a generic relation holds between the two SWTs, e.g., `Property1 hasDomain T1` and `Property1 hasRange T2` or inversely.
- **Sibling Relations:** when the two SWTs share a common ancestor, which can be either a direct or an indirect parent. Note that our definition covers the three sibling definitions described in [12].

[3] `http://wordnet.princeton.edu/`
[4] `http://tap.stanford.edu/data/`

– **Instance Of Relations:** such as `T1 instanceOf T2` or inversely. Unlike the previous relations, this relation is not considered by [12].

The identification of these relations can be made in two ways. First, a relation between SWT's might be declared **within a single ontology**. Second, if no single ontology mentions both SWT's, then a **cross-ontology relation discovery** can be performed by combining knowledge from several ontologies.

Cross-ontology relation discovery has been successfully implemented in the case of ontology matching [10]. An important issue to be considered is how to deal with potential contradictory relations, e.g., `T1 subClassOf T2` and `T1 disjointWith T2`. This remains a future work topic.

The semantically connected tags form the knowledge structures mentioned in the beginning of Section 2.1 and the tags not linked to SWTs or not related to other tags compose the set of uncovered tags. The study of the latter is expected to provide hints about how to evolve the Semantic Web, as described in Section 3. Next we describe the current implementation of our approach which identifies only subsumption and disjointness relations found in single ontologies.

2.2 Subsumption/Disjointness Discovery Based on One Ontology

The discovery of subsumption and disjointness relations between two terms within one ontology has been described and implemented on Swoogle'05 in [10]. Given two candidate concept names (`A` and `B`) as input, corresponding concepts are selected in online ontologies (`A'` and `B'`) by using strict string based anchoring. The possible semantic relations occurring between concepts in an ontology are shown using description logic syntax, e.g., `A'` $\sqsubseteq$ `B'` means that `A'` is a subconcept of `B'`. The returned relations are expressed with arrows, e.g., `A` $\xrightarrow{\sqsubseteq}$ `B`. The steps of this strategy in detail are:

1. Select ontologies containing concepts `A'` and `B'` corresponding to `A` and `B`;
2. If no such ontology is found, then `A` and `B` do not relate;
3. If there are returned ontologies, for each:
   - if `A'` $\equiv$ `B'` then derive `A` $\xrightarrow{\equiv}$ `B`;
   - if `A'` $\sqsubseteq$ `B'` then derive `A` $\xrightarrow{\sqsubseteq}$ `B`;
   - if `A'` $\sqsupseteq$ `B'` then derive `A` $\xrightarrow{\sqsupseteq}$ `B`;
   - if `A'` $\perp$ `B'` then derive `A` $\xrightarrow{\perp}$ `B`;

In a simple implementation we can rely on *direct* and *declared* relations between `A'` and `B'` in the selected ontology. But for better results *indirect* and *inferred* relations should also be exploited. For our experiments, we used an implementation relying on basic transitivity reasoning (i.e., taking into account all parents of `A'` and `B'`) and stopping as soon as a relation is found.

3 Experimental Results

The goal of our experiments is twofold. On the one hand, we wish to reveal how much of the semantic enrichment of folksonomy tags can already be automated by using the software developed in [10] which partially implements the current version of our envisioned algorithm (the part described in Section 2.2). On the other hand, we wish to understand any problematic issues so that they can be addressed in the design of the final, complete algorithm. At a higher level, these issues give an insight in how folksonomies and the Semantic Web relate. In a first experiment (Section 3.1) we applied the software developed in [10] to Flickr and Del.icio.us clusters generated by [12]. This experiment lead to valuable insights into issues that hamper the enrichment and prompted us to repeat the experiments with another set of clusters selected directly from Flickr. We discuss the second set of experiments in Section 3.2.

3.1 Experiment 1

The number of results obtained by running our algorithm with the clusters generated in [12] were surprisingly low. Two major reasons explain this. First, our implementation only searches for `subClassOf` and `disjointWith` relations. Unfortunately, the majority of tags in the clusters we work with are not related by these relations but by generic relations. The second major reason is that few of the tags in the analysed clusters could be identified in ontologies in the Semantic Web. Taking a closer look to the tags that were not found we individuated the following cases:

Novel terminology. Folksonomies are social artifacts, built by large masses of people and dynamically change to reflect the latest terminology in several domains. As such, they greatly differ from ontologies which are generaly developed by small groups of people and evolve much slower. Therefore, it is not surprising that many of the tags used in folksonomies, e.g., {`ajax, css`}, have not yet been integrated into ontologies. Identifying frequent folksonomy tags that are missing from ontologies has a great potential for the Semantic Web as it can provide the first step towards enriching existing ontologies with these novel terms.

Instances. When people tag resources, especially pictures, they more often tend to tag them with specific names rather than more abstract concepts. In particular, we frequently found names of people {`monica, luke, stephanie`}, names of places {`japan, california, italy`} and particular dates {`august2005, aug292005`}. Unfortunately, the current version of our system only works at terminological level (it deals only with concepts and not with ontology instances), so we did not identify any of these instances in the experiments. Apart from that limitation it is unlikely that instances related to people and specific dates can be reliably identified in ontologies anyway.

Photographic jargon. Given the scope of Flickr as a photo annotation and sharing site, many of the tags that are used reflect terms used in photography,

such as {`nikon, canon, d50, cameraphone, closeup, macro`}. Unfortunately, this domain is weakly covered in the Semantic Web.

Multilingual tags. Both Flickr and Del.icio.us (but especially Flickr) contain tags from a variety of languages and not only English. These tags are usually hard to find on the Semantic Web because the language coverage of the existing ontologies is rather low. Indeed, statistics[5] performed on a large collection of online ontologies (1177) in the context of the OntoSelect library indicate that 63% of these ontolgies contain English labels, while a much smaller percentage contains labels in other languages (German 13.25%, French 6.02%, Portuguese 3.61%, Spanish 3.01%).

Concatenated tags such as {`christmasornament, xmlhttprequest, librariesandlibrarians`} appear frequently but obviously it is hard to identify concepts with the same spelling.

Given the very low coverage of the Semantic Web for the above mentioned categories of tags, we decided to repeat the experiments for clusters of tags that are well-covered in the Semantic Web. Also, since at this stage our system only discovers subsumption and disjoint relations, we decided that the experiments should consider significantly larger clusters than those provided by [12].

3.2 Experiment 2

In the second set of experiments we relied on the lessons learnt from the first experiment to identify clusters of tags that would be more appropriate for our goal. To address the first conclusion (i.e., that clusters should be potentially well covered in the Semantic Web), we relied on the results of previous work in the context of ontology matching [10]. Follow up experiments revealed that domains related to food and animal species are well covered in the Semantic Web. Therefore, we selected a couple of tags from these domains, based on the concepts for which the most mappings were found during the matching experiments. We selected the tags: `mushroom`, `fruit`, `beverage` and `mammal`.

The next step was to identify clusters of tags related to each of these tags. As we said, we were looking for large clusters that would be more likely to accommodate subsumption relations and not just generic relations between tags. We chose the cluster generator provided by Flickr[6], since it returns much larger clusters of related tags than Del.icio.us and Technorati (moreover, since Del.icio.us and Technorati are mostly oriented towards news, business and web technologies, the clusters they provide for our tags in the food and animal domains are quite small).

The same algorithm as in Experiment 1 was then applied to these clusters. As expected, we found several relations among tags as depicted in the figures below (directed arrows represent `subClassOf` relations, dotted lines depict `disjointWith` relations). 23% of the investigated tags was discovered in

[5] `http://olp.dfki.de/OntoSelect/w/index.php?mode=stats`
[6] `http://www.flickr.com/services/api/flickr.tags.getRelated.html`

ontologies. Besides the tags between which we found relations, there were also sets of tags that could not be linked with any other tag in their cluster. We analyze these tag sets and describe possible causes that led to this failure.

The case of Mushroom. The semantic relations identified among the 21% of the tags related to `mushroom` by using online ontologies are depicted in Fig. 3. *Mushroom* was identified as a kind of *Fungi* and a kind of *Plant*. Also, we have learnt that it is disjunct with *Pizza*, *Pepper*, *Cheese* and *Tomato* and so are these with each other. *Mushroom* also co-occurs with *Soup*, *Rice* and *Onion*. As expected, there is no subsumption relation between these concepts and *Mushroom*. However, they are all subclasses of *Food*, as are *Tomato* and *Cheese* as well.

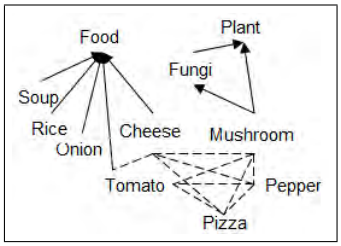


Fig. 3: Mushroom in the Semantic Web.

Type	Tags
Not covered by the SW	{`amanitamuscaria, toadstool, flyagaric`}
Generic relation (location)	{`nature, forest, garden, grass, moss`}
Generic relation (seasons)	{`autumn, fall, herfst`}
Generic relation (usage)	{`cooking, dinner, pasta, lunch`}
Colors	{`green, white, yellow`}
Photo jargon	{`macro, nikon, closeup`}

Table 1. `mushroom` related tags that could not be connected semantically

Table 1 shows some of the tags in the cluster of `mushroom` that could not be related semantically to any other tag, grouped according to the reason why they could not be linked. These are:

Tags that are not covered by the Semantic Web. These tags refer to kinds of mushrooms or scientific names that are not described in the Semantic Web. Generally, our experience is that currently very few online ontologies cover scientific labels.

Tags generically related to mushroom. The next three sets of tags are related to mushroom through other generic relations than subsumption or disjunction and describe locations, time and potential ways to use mushrooms.

Tags about colors. This set of tags is not surprising reflecting the fact that we retrieved the tag clusters from a photo-sharing system where users add color names to describe the image content of their photos. Note, however, that these colors might be meant to describe the rest of the tags associated to a resource, e.g., {`green pepper, white mushroom, yellow cheese`}. Unfortunately, because the creation of compound tags such as these is not well handled by folksonomies, users have to add each tag separately, thus loosing the relationship between them.

Photo jargon. The remaining group of tags are Flickr related tags, as we discussed in Experiment 1, and are not covered in the Semantic Web. Also, given the fact that they describe the photographs rather than their content, even if they were covered it is quite unlikely that they could be related to mushrooms or any other tag describing image content.

The case of Fruit We obtained interesting results for the cluster of `fruit` (Fig. 4) and the highest percentage of related tags, 29%. As fruits are well-covered by the Semantic Web, the generated semantic structure contains much more information than a single relation between the tags of the cluster. For example the multiple relations that exist between *Fruit* and *Vegetable*, and how this affects their common subclass, *Tomato*. In a biological context, a tomato is indeed the fruit of a tomato plant, however, normally one would classify tomatoes as types of vegetables. While such different views can co-exist, the fact that *Fruit* and *Vegetable* are disjoint makes this bit of knowledge inconsistent. Therefore, once such structures are derived from multiple ontologies, their consistency should be verified.

Also, according to online ontologies, *Fruit* is disjoint with *Dessert*. The validity of this statement depends on the point of view we adopt: some would argue that fruits are desserts, while others might consider desserts generally inappropriate catogorisation for fruits. Finally *Strawberry* and *Watermelon* were also found as subclasses of *Fruit*, but declaring them as subclasses of *Berry* and *Melon*, respectively, automatically infers they are also subclasses of *Fruit*.

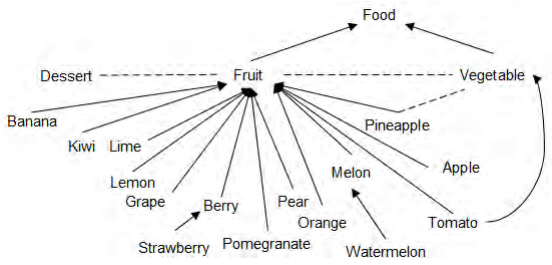


Fig. 4. *Fruit* in the Semantic Web

The tags that could not be connected to *Fruit* fall into five categories (see Table 2), two of which are related to colors and photo jargons, as discussed before. A new set of interesting tags describes attributes generally related to fruits: {`juicy, yummy, delicious, fresh, sweet`}. Unfortunately, most concepts in ontologies model nouns. Attributes are often modeled as properties (geneneric relations). Finally, the other two sets of interesting tags refer to fruit cultivation methods and possibly best seasons for consumption of specific fruits, which again share generic relations with fruits, currently not in the scope of our software.

Type	Tags
Attributes	{juicy, yummy, delicious, fresh, sweet}
Generic relation (cultivation)	{tree, nature, plant, seeds, leaves}
Generic relation (seasons)	{summer, autumn, fall, red, pink}
Colors	{brown, green, white, red, pink}
Photo jargon	{closeup, macro, canon}

Table 2. `fruit` related tags that could not be connected semantically

The case of Beverage. Beverage is the least covered tag with 18% of its related tags found to be connected in the Semantic Web. The knowledge structure that emerged from the semantic enrichment of the cluster related to `beverage` is shown in Fig. 5. As in the case of `fruit`, the cluster for `beverage` contains many concepts that were more specific than *Beverage*. Accordingly, these were identified to be in a subsumption relation with *Beverage* by our system. The two most interesting cases are of *White* being a subclass of *Beer* (white beer as a type of beer) and *Water* not being connected to *Liquid*. *Water*, though, was found to be related with *Fluid* which doesn't belong to the related tags of `beverage`. The tags that could not be related fall under the types of categories that we have already discussed in the previous cases and are presented in Table 3.

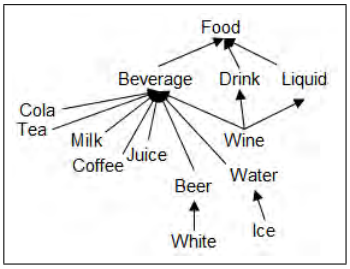


Fig. 5: Beverage in the Semantic Web.

Type	Tags
Not covered by the SW	{energy_drink, soda, martini, latte}
Generic relation (container)	{straw, mug, can, bottle, glass, cup}
Generic relation (event/place)	{breakfast, restaurant, party, starbucks}
Generic relation(ingredient)	{lemon, fruit, cream, orange}
Attributes	{hot, delicious, refreshing}
Colors	{brown, black, orange, green, red, pink}
Photo jargon	{closeup, macro, canon}

Table 3. `beverage` related tags that could not be connected semantically

Some types of beverages are not covered by the Semantic Web. It is interesting to note here that `latte` is not just an English word for a type of coffee, but also Italian for milk. The fact that it is not covered can be a side-effect of the low level of multilinguality in online ontologies, as we discussed in Experiment 1. Additionally, certain tags could be related to *Beverage* by generic relations, but these are not discovered by the current version of our system. These tags express types of containers, events and locations where beverages are served, as well as the ingredients of drinks. It is worth noticing that `orange` could belong both

to the categories representing colors and ingredients. The final set of tags that could not be related refer to attributes which, as discussed before, have generally a weak coverage on the Semantic Web.

The case of Mammal The last tag that was investigated is `mammal`. Relations for the 25% of its tags were found in the Semantic Web. Fig. 6 shows the structure derived from its cluster. It is interesting to observe that the subclasses of *Mammal* do not represent the same level of abstraction. We note many common names of animals like *Horse*, *Monkey*, *Rabbit*, but also two subclasses of higher abstraction, *Rodent* and *Feline*. This is another evidence that users annotate their content with a variable level of generality: although *Squirrel* and *Rabbit* appear in the graph as subclasses of *Mammal*, their superclass, *Rodent*, appears as well. This confirms the hypothesis put forward by [3] according to which different users will settle at different "basic levels" depending on their level of expertise.

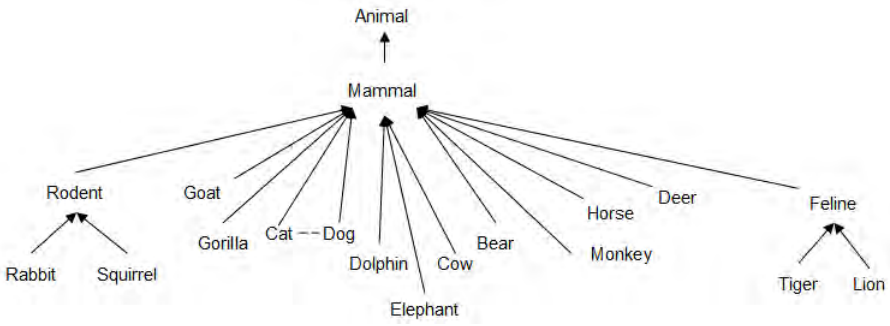


Fig. 6. *Mammal* in the Semantic Web

The tags that could not be related are displayed in Table 4. Most of these categories have been discussed previously, along with a set of tags that could have been related by generic relations indicating the location or habitat of mammals. Two tags were found to describe the state of the mammal when it was shot {`eating, sleeping`}. Finally, an interesting set of tags depicts body parts which should be related to mammals through a part-of relation.

Finally, it is worth pointing out that in all of the above here cases we identified certain tags, which were also found in Experiment 1, describing the places shown in the images, such as `barcelona`, `japan`, or the interests of the users, such as `ilovenature`, `stilllife` (we found 84.077 pictures annotated with `ilovenature` and 39.320 with `stilllife`).

Type	Tags
Not covered by the SW	{giraffe, seal, zebra}
Generic relation (location)	{zoo, nature, water, ocean, wild, farm, outdoors}
Generic relation (action)	{eating, sleeping}
Part-of	{fur, whiskers, eyes, face, nose}
Attributes	{cute, pet, funny, bunny}
Photo jargon	{portrait, closeup, macro, canon}

Table 4. `mammal` related tags that could not be connected semantically

4 Conclusions and Future Work

As an answer to our main research question, which is to explore whether folksonomies can be automatically enriched by harvesting the Semantic Web, based on the results of the preliminary experiments presented above, we can already conclude that it is indeed possible to automate the semantic enrichment of folksonomy tag spaces by harvesting online ontologies. By using these ontologies, we were able to automatically obtain semantic relations between the tags of several clusters of related tags. An immediate goal of our future work is to apply our approach on folksonomies and evaluate it in terms of Information Retrieval performance values (recall and precision). As an answer to our second research question, which is to identify the inherent characteristics of folksonomies and the Semantic Web and how they should be approached, the experiments also yielded relevant observations about these characteristics which have an impact on folksonomy enrichment process:

1. Folksonomy Characteristics. Our experiments show that many folksonomy tags fall in specific categories that require special attention. First, by being dynamically updated by large masses of people, folksonomies reflect the newest terminology within several domains (**novel terminology**). Second, many folksonomy tags refer to specific **instances** (names of people, places, dates). Third, folksonomies contain tags representing words in a variety of languages (**multilinguality**). Fourth, some of the tags that are frequently used depend on the purpose of the folksonomy and usually describe the resource itself rather than its content (**folksonomy jargon**). Fifth, folksonomy tags often describe **attributes** of the content, for example, colors (especially in Flickr). Sixth, there are many **concatenated tags** which describe a large number of photographs and need to be exploited. Finally, a **broad range of semantic relations** can exist between tags, including subsumption, disjointness, meronymy and many generic relations (e.g., location).

2. Semantic Web Characteristics. The most important observation regarding the Semantic Web is that even if it is growing fast it still suffers from *knowledge sparseness* (i.e., it presents good coverage for certain topics, but very low coverage for others). Due to this limitation, we needed to restrict our experiments to domains that are well-covered (related to animals and food). Also,

some of the categories of tags that appear frequently in folksonomies are difficult to find in online ontologies. First, **novel terminology** that emerges from folksonomies is often missing from ontologies. Second, the majority of **specific instances** that appear in folksonomies cannot be found (e.g., `aug2004`) or are difficult to reliably map to ontology instances (e.g., `monica`). Place names are an exception to this. Third, few of the online ontologies contain **multilingual labels**, therefore tags in languages other than English are unlikely to be found in ontologies. Fourth, **specific jargons**, such as those related to photography are weakly covered as well. Fifth, online ontologies are rather **poor in describing generic attributes** such as color. One of the reason for this is that attributes are most often modeled as part of properties rather than concepts.

We are confident, however, that surpassing some of the current limitations is a matter of time as many of them will be solved as more ontologies will appear online. For example, the AGROVOC[7] ontology contains roughly 16000 concepts and their labels in 12 different languages. Making this single ontology available online will positively impact on the issue of anchoring multilingual tags. Nevertheless the appearance of more online ontologies can also be seen as a potential risk for this work as different ontologies reflect different views which often lead to contradictory bits of knowledge. Combining these bits may result in inconsistencies in the derived semantic structures. However, existing reasoning techniques can be used to filter out and eliminate possible inconsistencies.

Being aware of these characteristics help us to identify the **current limitations of our software**. Our software only implements a subset of the functionality envisioned for the enrichment algorithm. First, it is currently implemented on Swoogle'05 which lags behind in ontological content. Our final algorithm will be built on top of up-to-date semantic search engines [1]. Second, the anchoring mechanism is based on strict string matching and therefore needs to be extended to more flexible anchoring. Third, from the broad range of semantic relations that can exist between tags, our software only identifies subsumption and disjointness. Obviously, extensions are needed that can discover the other types of relations as well. Finally, note that we have only experimented with finding relations within a single ontology and excluded cases when knowledge can be derived by combining facts from multiple ontologies. Another important future work will be to implement this cross-ontology relation derivation.

The experimental work reported in this paper indicates that the proposed enrichment process has the potential to benefit both folksonomies and the Semantic Web, thus answering our third research question. On the one hand, even using a software with limited functionality we were able to derive *explicit* semantic relations between tags, thus going beyond existing methods that identify *implicitly* inter-related tags. We believe this could considerably enhance content retrieval in folksonomies. On the other hand, the differences between folksonomies and ontologies (such as novel terminologies emerging in several languages) can be used to evolve the Semantic Web. This valuable knowledge available in folksonomies could allow keeping online ontologies up to date, extending them with

[7] `http://www.fao.org/agrovoc`

multi-lingual information and evolving them towards being truly *shared* conceptualisations of a much broader range of domains.

Acknowledgements

We thank M. d'Aquin for his useful comments on earlier versions of this paper. This work was funded by the Open Knowledge and NeOn projects sponsored under EC grant numbers IST-FF6-027253 and IST-FF6-027595.

References

1. M. dAquin, M. Sabou, M. Dzbor, C. Baldassarre, L. Gridinoc, S. Angeletou, and E. Motta. Watson: A gateway for the semantic web. In *Poster Session at ESWC'07*, 2007.
2. G.Begelman, P. Keller, and F.Smadja. Automated Tag Clustering: Improving search and exploration in the tag space. In *Proc. of the Collaborative Web Tagging Workshop at WWW'06*, 2006.
3. S. Golder and B.A. Huberman. The Structure of Collaborative Tagging Systems. HPL Technical Report, 2005.
4. J. Gracia, R. Trillo, M. Espinoza, and E. Mena. Querying the Web: A Multiontology Disambiguation Method. In *Proc. of ICWE'06*, 2006.
5. A. Hotho, R. Jaschke, C. Schmitz, and G. Stumme. Information Retrieval in Folksonomies: Search and Ranking. In *Proc. of ESWC'06*, 2006.
6. L.Ding, T. Finin, A. Joshi, R. Pan, R. S. Cost, Y. Peng, P. Reddivari, V.C. Doshi, and J. Sachs. Swoogle: A Search and Metadata Engine for the Semantic Web. In *Proc. of the 13th ACM Conf. on Information and Knowledge Management*, 2004.
7. P. Mika. Ontologies are us: A unified model of social networks and semantics. In *Proc. of ISWC'05*, 2005.
8. P.A. Pantel. *Clustering by Committee*. PhD thesis, 2003.
9. E. Peterson. Beneath the Metadata: Some Philosophical Problems with Folksonomy. *D-Lib Magazine*, 12(11), November 2006.
10. M. Sabou, M. d'Aquin, and E. Motta. Using the Semantic Web as Background Knowledge for Ontology Mapping. In *Proc. of the Int. Workshop on Ontology Matching (OM-2006)*, 2006.
11. P. Schmitz. Inducing Ontology from Flickr Tags. In *Proc. of the Collaborative Web Tagging Workshop at WWW'06*, 2006.
12. L. Specia and E. Motta. Integrating Folksonomies with the Semantic Web. In *Proc. of ESWC'07*, 2007.
13. Thomas Vander Wal. Folksonomy coinage and definition. 2007.
14. H. Wu, M. Zubair, and K. Maly. Harvesting Social Knowledge from Folksonomies. In *In Proc. of HYPERTEXT '06*, 2006.
15. X. Wu, L. Zhang, and Y. Yu. Exploring Social Annotations for the Semantic Web. In *Proc. of WWW'06*, 2006.

Computing Word-of-Mouth Trust Relationships in Social Networks from Semantic Web and Web2.0 Data Sources

Tom Heath[1], Enrico Motta[1], Marian Petre[2]

[1]Knowledge Media Institute and Centre for Research in Computing
[2]Department of Computing and Centre for Research in Computing
The Open University, Walton Hall, Milton Keynes, MK7 6AA, United Kingdom
{t.heath, e.motta, m.petre}@open.ac.uk

Abstract. Social networks can serve as both a rich source of new information and as a filter to identify the information most relevant to our specific needs. In this paper we present a methodology and algorithms that, by exploiting existing Semantic Web and Web2.0 data sources, help individuals identify *who in their social network knows what*, and *who is the most trustworthy source of information on that topic*. Our approach improves upon previous work in a number of ways, such as incorporating topic-specific rather than global trust metrics. This is achieved by generating *topic experience* profiles for each network member, based on data from *Revyu* and *del.icio.us*, to indicate who knows what. Identification of the most trustworthy sources is enabled by a rich trust model of information and recommendation seeking in social networks. Reviews and ratings created on *Revyu* provide source data for algorithms that generate *topic expertise* and *person to person affinity* metrics. Combining these metrics, we are implementing a user-oriented application for searching and automated ranking of information sources within social networks.

1 Introduction

Social networks can serve as both a rich source of new information and as a filter to identify the information most relevant to our specific needs. Making optimal use of the knowledge within our social networks requires that we know firstly *who knows what*, and secondly *who is the most appropriate source of information on that topic*. In this paper we present a methodology and algorithms that address these issues by exploiting existing Semantic Web and Web2.0 data sources. Our approach supports an application that helps the user identify which members of their social networks may have knowledge on a particular topic, and of which topics each member of their network has knowledge. This is achieved by generating *topic-experience* profiles for each known person based on data from *Revyu* [4] reviews and ratings, and *del.icio.us*[1] social bookmarks.

[1] http://del.icio.us/

The second requirement is addressed by a rich trust model of information and recommendation seeking in social networks, based on previous empirical research. Reviews and ratings created on *Revyu* provide source data for algorithms that generate *topic expertise* and *person to person affinity* metrics. Combining all metrics derived in this fashion, we are implementing a user-oriented application for searching and automated ranking of information sources within social networks.

This paper describes in detail our methodology and algorithms for computing trust relationships, and briefly outlines the application we are developing that makes use of them. After reviewing related work in Section 2, Section 3 outlines the advantages of our approach. In Section 4 we summarize the findings of a previous study into how people choose sources for word of mouth recommendations. Section 5 introduces our technical approach, whilst Section 6 describes algorithms we have developed for computing trust relationships in word of mouth recommendation seeking scenarios, based on the findings of the previous study. Section 7 gives an overview of how these metrics are being used in applications that support information seeking using trust relationships in social networks. Section 8 concludes the paper with an outline of future work.

2 Related Work

The work of Granovetter [1] highlighted how social networks can serve as a source of new information to which an individual may not otherwise have access. In the context of job hunting, he found that weak, rather than strong, social ties are particularly useful, in that they are sufficiently well connected outside of the individual's immediate network (i.e. a sufficient proportion of acquaintances were not shared) as to provide valuable access to otherwise unavailable information about job opportunities.

In addition to this role of information source, our social networks can also serve as a filter, helping us identify the most relevant or appropriate information. At least two factors underpin this: firstly, the principle of homophily [5] states that we are likely to have more in common with members of our social networks than with other members of the population, and more likely to like what they like; secondly, we are better able to judge the appropriateness and trustworthiness (as information sources) of people we know, as we have greater background knowledge of their competence and trustworthiness in a particular domain.

These processes may be assisted by Web technologies in a number of ways. *Collaborative filtering* [6] recommender systems such as GroupLens [7] have typically sought to assist in information filtering by identifying others that share our preferences for newsgroup postings or some other type of item (such as items in an e-commerce site). Variations such as *Amazon* recommendations [8] perform a similar function but instead correlate item rather than people profiles. In the person-to-person approaches, collaborative filtering creates for each of us a social network of unknown others who nevertheless have shared tastes, and through whose preferences information can be filtered on our behalf. Whilst this can be of great value in informing decision-making, it does not allow us to use our own knowledge in

assessing the relevance or trustworthiness of a source, and does not address situations where we require recommendations from domain experts, irrespective of their likeness to ourselves.

So how do people determine the trustworthiness, as information sources, of the people in their social networks? Various studies of information seeking in workplace settings [2, 3] found that people decide whom to ask for information based on what they know of the person, and how they value their knowledge and skills. Both studies found an effect of perceived source quality in determining the likelihood that an individual asks another for information.

In previous work [9] we extended these findings beyond workplace settings, and refined the notion of source quality or trustworthiness. These findings are summarized in Section 4 below. In this paper we will report on how we are using Semantic Web and Web2.0 data sources and social networks to calculate trust ratings between individuals, and how we are using these ratings to support information seeking from known and trusted sources.

Some existing work has been carried out in this area. For example, Massa and Avesani [10] use trust propagation mechanisms to increase the coverage of recommender systems without sacrificing the quality of recommendations to users. Perhaps the best known work in this area from a specifically Semantic Web perspective is that of Golbeck and colleagues. Golbeck and Mannes [11] use manual trust annotations between people (on a 1 to 10 scale) combined with provenance information about trust ratings and social network connections to infer trust ratings between unknown sources. Whilst this can be of value where insufficient annotations are provided by one's social network, it suffers a number of limitations. Firstly the trust ratings (either manual or computed) are not topic-specific; users are required to make global statements of their trust in another person, without further context being provided. This approach also requires sufficient manual trust annotations to bootstrap the process, without being able to rely on existing sources of information. In contrast, our approach aims to compute *person-person* and *person-topic* trust ratings according to a richer model of trust in word of mouth recommendation seeking, and based on existing data sources available in the Web.

3 Our Approach: Trusted Recommendations from a Social Network

We are investigating the use of social networks to provide relevant information and recommendations. In contrast to existing work, our approach aims to identify trusted sources from among known members of one's social network. This follows the principle that knowing the right person to ask is often the greatest challenge in seeking information or recommendations.

This *known person*, *source-centric* (rather than item-centric) approach has a number of advantages. It allows the user to employ existing knowledge of their social network to assess the quality and impartiality of recommendation sources, and follow-up enquiries with the source as they see fit. Therefore, in contrast to collaborative filtering our approach is less vulnerable to spamming, for the simple

reason that each user's exposure to the recommendations of others is limited in the first instance to those people they know. We proceed on the assumption that most users are unlikely to know others who wish to manipulate search indices on an ongoing, systematic basis. A recent investigation [12] (albeit journalistic, rather than scientific) demonstrated how easily ratings on travel review and recommendation sites such as *TripAdvisor*[2] can be skewed by those with a vested interest in promoting a particular establishment. Personally knowing those providing a review or recommendation acts as a safeguard against this form of manipulation.

Secondly, our source-centric approach does not assume completeness of the information in the system. For example, for a conventional recommender system to be able to recommend a hotel in Madrid to User A, some record of a hotel in Madrid must exist in the system. In contrast, whilst our approach can identify specific instances of recommended hotels in Madrid, simply identifying those known people with some knowledge of Madrid is sufficient to begin answering the user's information needs, without requiring substantial amounts of information. This is analogous to simply asking "who do I know that knows anything about Madrid?", and is in contrast to conventional collaborative filtering approaches, that whilst they may list "people like you", they are generally aimed towards informing the user that "people like you also liked X". In this sense they are item- rather than source-centric.

Thirdly, Linden, Smith, and York [8] outline limitations of traditional collaborative filtering that stem from its computational expense over large datasets. Computing the *co-preference*[3] between all users of a system has been found not to scale where large numbers of users are concerned. By constraining recommendations to those coming from members of a user's social network, we reduce the number of co-preference relationships that must be computed in the system. We anticipate that such an architecture will allow the system to scale more readily.

Lastly, by using Semantic Web technologies we are able to exploit and integrate data from many different sources in computing trust relationships. Our approach uses FOAF-based definitions of users' social networks [13], allowing "friend lists" built up across different services to be reused. *Revyu* provides data about reviews and ratings in crawlable RDF and via a SPARQL endpoint. This brings practical benefits during development (such as query flexibility, ability to reuse common libraries) compared to the more restrictive data access allowed by *del.icio.us*. Crucially however, by being Semantic Web-aware, our approach allows for the generation or refinement of trust ratings based on additional Semantic Web data sources as they become available. This issue is discussed in Section 8.

4 Previous Findings: Trust in Recommendation Seeking

In a previous paper [9] we presented the results of an empirical study examining how people select recommendation sources from among their social networks, and the factors that influence these decisions. Participants were presented with four recommendation seeking scenarios, asked to explain from whom they would seek

[2] http://www.tripadvisor.com/

[3] The degree of preference two individuals share for an item

recommendations in each scenario, and to explain their reasons for these choices. Analysis of participants' responses identified five factors underlying the trust or confidence participants had in recommendations from specific sources: the *expertise*[4], *experience*[5], and *impartiality*[6] of the source with regard to the topic of the recommendation seeking, the *affinity*[7] between the source and recommendation seeker, and the *track record*[8] of previous recommendations from the source.

These trust factors varied in their frequency of occurrence in participants explanations for choosing a particular source. *Expertise*, *experience*, and *affinity* occurred most frequently, with relatively low occurrences of the *impartiality* and *track record* factors. Furthermore, the emphasis given to each of these factors was found to vary according to the characteristics of the recommendation seeking task.

Results suggested that the *criticality* of the task and the *subjectivity* of possible solutions were of primary importance in determining which trust factors were emphasised. In scenarios seen by participants as more critical, greater emphasis was placed on the recommendation source having relevant *expertise*. In contrast, in scenarios in which potential solutions were seen as more subjective, participants placed greater evidence on sources with which they shared a strong affinity.

A major shortcoming of the work of Golbeck and Mannes [11] is that trust relationships are represented as global traits between users, rather than being topical or domain-specific. A foundation for our work is the principle that trust can be topical, in that one person may be highly trusted for recommendations in one domain but trusted very little in others. For example, one may trust a friend who works in banking to give sound financial advice, but never trust her film recommendations. The findings of our previous study support our assertion of trust topicality, and suggest that any robust model of trust in word of mouth recommendation must take this into account.

It is worth noting that whilst the factors expertise, experience, and impartiality were clearly domain specific and therefore topical in nature, the study did not give a strong indication of affinity as a topical factor, but rather as a global construct. The range of responses that informed the *affinity* factor suggests that it represents more than simply shared tastes, encompassing instead similar outlooks on life, values, and expectations: "I would ask X, because we see the world in the same way".

5 Computing Knowledge and Trust Relationships

Based on the trust factors identified in this previous study, we have developed algorithms for computing *people-people* and *people-topic* trust metrics that signify

[4] The source has relevant expertise, which may be formally validated through qualifications or acquired over time

[5] The source has experience of solving similar scenarios, but without extensive expertise

[6] The source does not have vested interests in a particular resolution to the scenario

[7] The source has characteristics in common with the recommendation seeker such as shared tastes, standards, viewpoints, interests, or expectations

[8] The source has previously provided successful recommendations to the recommendation seeker

respectively the affinity-based trust relationship between two individuals, and the expertise- and experience-based trustworthiness of an individual with regards to a topic. The metrics generated by these algorithms provide the foundations on which our system is built. An overview of the system is provided in Section 7.

We argue that auto-generating trust metrics from existing background data sources is crucial, for a number of reasons. Firstly, such an approach can help overcome the bootstrapping/cold-start problem, whereby a system is only useful to the user once they have provided a certain amount of data specifically to that system. We are exploiting a range of existing and widely used Web2.0 data sources, such as del.icio.us and Flickr, in the generation of our *experience* trust metrics. Initial weak metrics generated from these sources are then enhanced based on richer data from our *Revyu* Semantic Web reviewing and rating site. The integration of further sources into the trust metric generation process is technically feasible and highly desirable. Secondly, reuse of existing sources lessens the burden on the user, as they need not provide new data about their preferences to our system. Instead they can immediately reap the benefits of data they have provided in one system (such as bookmarks in del.icio.us, or reviews in Revyu), in the form of enhanced search results and personalization in our system.

Lastly, one additional mechanism for determining the trustworthiness of people's recommendations in a domain would be to ask them to rate their knowledge or expertise in a number of domains. However, such an approach would require a comprehensive yet manageable list of topics or domains, which by definition scales poorly to the full range of topics on which users might require recommendations. By reusing data from external sources that are themselves unconstrained in their coverage of topics (as users can use any tags they wish), we are not constraining the domains or topics in which trust metrics can be calculated.

In computing trust metrics for use within our system, we have given priority to the three trust factors arising most frequently in our previous study: *expertise*, *experience*, and *affinity*. Developing algorithms that directly represent the trust factors has not been possible in all cases. In particular, computing an expertise score in any one domain is problematic, as appropriate sources of background knowledge that indicate expertise are not widely available on the Web, are widely dispersed by topic, and are not generally available in structured, machine-readable form. For example, one's family doctor may have expertise in general healthcare. However, evidence of this in the form of a machine-readable certificate of qualification and competence from a recognised medical authority is not available on the Web. Consequently we have developed a metric (called *credibility*) that serves as a proxy for expertise. An individual is deemed credible with respect to a particular topic if their ratings of items related to that topic correlate highly with those of the community as a whole.

Similarly, large volumes of data are available on the Web that may indicate an individual's experience with regard to a particular topic. However, automatically validating with any degree of confidence that this is the case may not be feasible. Therefore a proxy metric (*usage*) has been developed that suggests an individual has experience in a particular topic. Comparing ratings between individuals allows us to compute affinity metrics with some degree of confidence, without resorting to proxy measures.

6 Algorithms for Generating Trust Metrics

The algorithms used to compute trust metrics in our system are detailed below. The algorithms rely primarily on data from Revyu, however *usage* (*experience*) metrics are also computed based on del.icio.us tagging data. Tags used in Revyu and del.icio.us seed the list of topics for which individuals may have usage or credibility scores. In Section 8 we discuss further potential Semantic Web data sources on which to base trust calculations.

6.1 Credibility (Expertise) Algorithm

```
for each tag in Revyu
    get all items tagged with that tag, by anyone
        for each item
            find the mean item rating
            for each review of the item
                subtract rating from mean rating to
                give a rating distance
                adjust sign of the rating distance to
                ensure it is positive
                divide rating distance by highest
                possible rating minus 1 to give
                normalized rating distance
                subtract normalized rating distance
                from 1 to give credibility score for
                that review in the range 0-1
                sum each reviewer's credibility
                scores for the current tag to give a
                credibility total for this tag
for each reviewer with a credibility total for this tag
    divide the credibility total by the number of
    reviews from which it is gained, giving a
    reviewer's credibility score for that tag, in the
    range 0-1
```

Fig. 1. Credibility (Expertise) algorithm in pseudo-code

At present the algorithm does not take into account tags for which only one item exists, or tags for which multiple items exist but where all have only been reviewed by the same person. This can lead to the situation where an individual is assigned a

credibility rating of 1 for a particular topic, by virtue of being the only reviewer of things tagged with that tag. It could be argued that within the scope of the knowledge currently held within the system, this person is justifiably credible and expert on the topic, as no contradictory information exists. However, we do not accept this argument, and anticipate some negative effects of this artifact when we evaluate the algorithms. Methods for mediating this effect are being sought in ongoing research.

6.2 Usage (Experience) Algorithm

This algorithm calculates the prevalence of an individual in the reviews of items that have been tagged with a particular tag, thereby providing a relative measure of their experience with the topic.

```
for each tag in Revyu
    count how many times each reviewer has reviewed an
    item tagged with that tag (by anyone); this gives a
    reviewer's tag count
    find the highest of these tag counts
divide each reviewer's tag count by the highest tag
count to give a usage score in the range 0-1
```

Fig. 2. Usage (Experience) algorithm in pseudo-code

Catching all people who have reviewed something that has ever been tagged with the target tag helps ensure that people are credited with experience in a relevant domain, even if they haven't used a particular keyword tag themselves. This helps ensure a broader spread of experience scores across related topics.

One consequence of this algorithm is that the individual with the highest *tag count* will be assigned a usage (experience) score of 1 for that topic, by virtue of having reviewed the greatest number of things tagged with a particular tag, and irrespective of the overall number of reviews of items tagged with that tag. Following evaluation we may modify this algorithm to ensure no scores of 1 can be assigned, and also to adjust scores relative to the total number of reviews.

6.3 Affinity Algorithm

The following algorithm computes an affinity score between an individual and another person they know, based on analysis of their reviews in Revyu. In addition to Revyu review data, the algorithm must be seeded with some basic details of the known person. This is supplied to the algorithm in the form of a FOAF description of the user's social network.

```
get all reviews by the user (User A)

get all reviews by the known person (User B)

count the number of items that both users have reviewed

divide this by the highest number of total reviews by
either user, to give an item overlap ratio in the range
0-1

where both users have reviewed the same item

          subtract the rating of User B from that of
          User A, to give a rating distance

          adjust the sign of the rating distance to
          ensure it is positive

          divide rating distance by highest possible
          rating minus 1, to give a normalized rating
          distance in the range 0-1

          subtract the normalized rating distance from 1
          to give a rating overlap for that review

          sum all item-level rating overlaps between
          users A and B, then divide by the number of
          items that both users have reviewed, to give a
          mean rating overlap

combine the item overlap ratio and mean rating overlap
to produce a measure of the affinity between User A and
User B
```

Fig. 3. Affinity algorithm in pseudo-code

At present several aspects of the affinity computation process are subject to variation pending the outcome of evaluations into the effectiveness of the algorithms. Firstly, the relative importance of *item overlap ratio* and *mean rating overlap* in computing affinity is not fully clear, and may vary according to the item overlap ratio. For example, a high mean rating overlap based on few overlapping items may be of less value as a measure of affinity than a slightly lower mean rating overlap based on a large number of overlapping items. The most reliable means for combining these measures is an ongoing question for our research. One option may be to base affinity scores purely on *mean rating overlap*, weighted according to the number of overlapping items. An alternative may be to introduce confidence measures whereby affinity scores are based solely on mean rating overlap, but the confidence of this measure is expressed based on the item overlap ratio.

6.4 Generating Usage (Experience) Scores from del.icio.us Data

In order to increase the range of topics for which users in the system have usage/experience scores, we have extended the *usage* (experience) algorithm to take into account users' tags on *del.icio.us*. Where a user of the system has a *del.icio.us* account, their most used tags are retrieved. For each tag that has received a certain amount of usage (above an arbitrary threshold), the user is recorded as having some *experience* of that topic. A standard nominal experience score (currently 0.1) is assigned irrespective of the frequency of usage of the tag above the threshold, in recognition that tag usage is not necessarily strongly correlated with real experience of the topic. For example, in the course of researching possible holiday destinations a user may bookmark many resources using the tag *hawaii*, but eventually choose Mexico instead for their holiday. In contrast, where a user has reviewed an item we can be reasonably confident that they have some experience of the topics denoted by that item's tags.

Where a user has an existing experience score for a particular topic that exceeds the nominal score derived from their del.icio.us tags, the existing score stands unchanged. Where they have an existing score lower than the nominal score, this is increased in line with the nominal score for del.icio.us-derived experience. No attempt is made to supplement Revyu-derived *credibility* and *affinity* metrics based on del.icio.us data, as bookmarks do not carry ratings, endorsements, or other value judgments from which these may be derived.

6.5 Representing Computed Trust Relationships

Once computed, trust relationships based on these metrics are stored in a triplestore, according to a simple ontology that models the relationships between people and topics identified in our earlier study. This triplestore provides the data for the application outlined below. Trust relationships will also be republished on the Web for potential reuse in other applications.

7 Supporting Information Seeking with Trusted Social Networks

Using trust relationship data computed according to the algorithms detailed above, we are currently completing the implementation of a system that enables people to locate and explore trusted information sources within their social networks, and access items rated highly by these sources. An example of output from the system is shown in Figure 4 below.

Fig. 4. System output showing *film* experts in the first author's social network, ranked according to *expertise*

As discussed above, the role of trust in information seeking is not constant, but varied and situational, depending on characteristics of the task such as its *criticality* and *subjectivity*. Consequently, in our approach the relative importance of *topic expertise* and *person to person affinity* in ranking of potential information sources is varied according to the *criticality* and *subjectivity* of the information seeking task. We intend to carry out user evaluations to assess the relative merits of different mechanisms for representing *criticality* and *subjectivity* in the system. Current approaches being considered include allowing the user to select criticality and subjectivity measures in the interface, and pre-categorizing the domains of queries according to their *criticality* and *subjectivity* profiles.

8 Conclusions and Future Work

In this paper we have presented our approach to generating trust profiles for members of a user's social network, in the context of word of mouth recommendation seeking. This approach is based on algorithms for computing *person-topic* (expertise, epxerience) and *person-person* (affinity) trust metrics, that have been developed based on previous research. By utilizing people's social networks, and employing a rich model of trust in recommendation seeking, our approach overcomes the limitations of previous work in the field.

In addition to completing implementation of the system outlined above, a number of outstanding issues remain which are the subject of ongoing research. Firstly we are investigating the integration of additional sources of data. The contents of users' FOAF files, when combined with other Semantic Web datasets, provide a potentially rich source of information about users' experience of particular topics. For example, where a user states in their FOAF file that they are `based_near` a particular location, we can assume they have some experience of this location, and consequently increase their experience rating for this topic. Use of the Geonames service[9] may allow us to locate other nearby locations, and assume the user also has some (although likely less) experience of these.

Amongst Web2.0 data sources, *Flickr*[10] in particular may provide a good basis for assessing people experience of particular locations or activities, as photos are likely to be tagged with a location name. In contrast however, it may also lead to significant noise in the system where people have tagged items using words that whilst representing some aspects of the contents of the picture, do not indicate particular experience of a topic. Whilst sources of reviews such as Amazon and Yahoo Reviews are potentially rich in terms of quantity of reviews, they do not provide information from known sources, as reviewers are rarely reliably identifiable.

Regarding the trust relationship algorithms, we aim to investigate how trust relationships may decay over time, and how any rate of decay may vary across different domains. For example, the trustworthiness of a person as a source of knowledge on ancient history may decay very slowly, whereas trust in another individual as a source of restaurant recommendations in London may quickly decay if it isn't regularly updated. Representing these issues in our algorithms is an area of future investigations.

Lastly we aim to use patterns in tag co-occurrence to disambiguate topics, and also as a means to propagate trust scores in one topic to others that are related. Throughout these processes we will continue to evaluate the techniques we develop to ensure that they reliably address user needs.

[9] http://www.geonames.org/
[10] http://flickr.com/

Acknowledgements

This research was partially supported by the Advanced Knowledge Technologies (AKT) project. AKT is an Interdisciplinary Research Collaboration (IRC), which is sponsored by the UK Engineering and Physical Sciences Research Council under grant number GR/N15764/01. The AKT IRC comprises the Universities of Aberdeen, Edinburgh, Sheffield, Southampton and the Open University.

References

1. Granovetter, M. S.: The Strength of Weak Ties. The American Journal of Sociology 78 (1973) 1360-1380
2. Borgatti, S. P., Cross, R.: A Relational View of Information Seeking and Learning in Social Networks. Management Science 49 (2003) 432-445
3. O'Reilly, C. A.: Variations in Decision Makers' Use of Information Sources: The Impact of Quality and Accessibility of Information. The Academy of Management Journal 25 (1982) 756-771
4. Heath, T., Motta, E.: Reviews and Ratings on the Semantic Web. In: Proc. Poster Track. 5th International Semantic Web Conference (ISWC2006) (2006)
5. McPherson, M., Smith-Lovin, L., Cook, J. M.: Birds of a Feather: Homophily in Social Networks. Annual Review of Sociology 27 (2001) 415–44
6. Goldberg, D., Nichols, D., Oki, B. M., Terry, D.: Using Collaborative Filtering to Weave an Information Tapestry. Communications of the ACM 35 (1992) 61-70
7. Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R., Riedl, J.: GroupLens: applying collaborative filtering to Usenet news. Communications of the ACM 40 (1997) 77-87
8. Linden, G., Smith, B., York, J.: Amazon.com Recommendations: Item-to-Item Collaborative Filtering. IEEE Data Engineering Bulletin (2003) 76-80
9. Heath, T., Motta, E., Petre, M.: Person to Person Trust Factors in Word of Mouth Recommendation. In: Proc. CHI2006 Workshop on Reinventing Trust, Collaboration, and Compliance in Social Systems (Reinvent06) (2006)
10. Massa, P., Avesani, P.: Trust-aware Collaborative Filtering for Recommender Systems. In: Proc. International Conference on Cooperative Information Systems (CoopIS) (2004)
11. Golbeck, J., Mannes, A.: Using Trust and Provenance for Content Filtering on the Semantic Web. In: Proc. WWW2006 Workshop on Models of Trust for the Web (2006)
12. Walsh, G., Swinford, S.: Hotel review websites: a five-star scam. (2006) http://travel.timesonline.co.uk/tol/life_and_style/travel/article634136.ece
13. Brickley, D., Miller, L.: FOAF Vocabulary Specification.http://xmlns.com/foaf/0.1/ (2005)

FolksOntology: An Integrated Approach for Turning Folksonomies into Ontologies

Céline Van Damme[1], Martin Hepp[2], and Katharina Siorpaes[2]

[1]Vakgroep MOSI, Vrije Universiteit Brussel, Brussels, Belgium
[2]Digital Enterprise Research Institute (DERI), University of Innsbruck, Innsbruck, Austria
celine.van.damme@vub.ac.be, mhepp@computer.org,
katharina.siorpaes@deri.org

Abstract. We can observe that the amount of non-toy domain ontologies is still very limited for many areas of interest. In contrast, folksonomies are widely in use for (1) tagging Web pages (e.g. del.icio.us), (2) annotating pictures (e.g. flickr), or (3) classifying scholarly publications (e.g. bibsonomy). However, such folksonomies cannot offer the expressivity of ontologies, and the respective tags often lack a context-independent and intersubjective definition of meaning. Also, folksonomies and other unsupervised vocabularies frequently suffer from inconsistencies and redundancies. In this paper, we argue that the social interaction manifested in folksonomies and in their usage should be exploited for building and maintaining ontologies. Then, we sketch a comprehensive approach for deriving ontologies from folksonomies by integrating multiple resources and techniques. In detail, we suggest combining (1) the statistical analysis of folksonomies, associated usage data, and their implicit social networks, (2) online lexical resources like dictionaries, Wordnet, Google and Wikipedia, (3) ontologies and Semantic Web resources, (4) ontology mapping and matching approaches, and (5) functionality that helps human actors in achieving and maintaining consensus over ontology element suggestions resulting from the preceding steps.

1. Introduction

It has been argued e.g. in [1] that the insufficient involvement of users in the construction of ontologies is a significant cause for the current shortage of and the unsatisfying coverage found in domain ontologies. One of the reasons for this deficiency is that there are high barriers for laymen users for suggesting new conceptual elements. For example, a new concept, instance or property is added to the ontology only by a privileged group. This requires that ontology users with domain expertise take the burden and have the skills to make respective suggestions, which is different from the evolution of a natural language, where a new word can be invented on the spot when needed and immediately added to the vocabulary [1, 2].

Also, since ontology specifications are expressed in a formal language, potential users face difficulties in understanding the formal specifications of the ontology [1, 2]. This is important, since the inferences authorized by using a given ontology are represented only in its formal semantics, i.e. to what one commits to when adopting a particular ontology is not obvious from the human-readable labels of ontology elements but only from the associated axioms. In addition to that, we can observe that the detachment of ontology *usage* (e.g. creating annotations) from ontology

construction and maintenance in current practice cuts off valuable feedback and actually makes the social agreement over ontology elements brittle and vague.

Tagging, i.e., users describing objects with freely chosen keywords (tags) in order to retrieve content more easily, avoids these limitations, since new tags can be introduced on the spot when needed and the construction and maintenance of the tags is closely linked to their actual usage.

While the resulting tag sets and their assignment to objects are at first only reflecting subjective conceptualizations, many of those *subjective* representations can be used to derive *intersubjective* representations. Such aggregation of raw tag data leads to a flat bottom-up categorization or folksonomy [3]. Popular examples of the tagging/folksonomy mechanism are found in the social bookmark manager deli.cio.us (http://del.icio.us), the image sharing system Flickr (http://www.flickr.com), and the blog search engine Technorati (http://technorati.com).

Tagging features create a wealth of data that reflects (1) subjective assignments between words and categories of objects, (2) intersubjective patterns in these associations, and (3) implicit information on social networks.

However, tags are flat and no relationships or conceptual meanings are formally attached to them. This causes problems such as (1) lexical ambiguity; for instance, the tag “bank” can mean a financial institution or it can be used in the context of a river edge; (2) different tags (e.g. “NY” and “big_apple”) may refer to the same concept (e.g. the city New York), and (3) specialized (e.g. “seagull”) and more general tags (e.g. “bird”) may be attributed to the same object (e.g. a picture of a seagull on Flickr) [4].

Also, the same tag may be used for very different objects in clearly distinct contexts. For example, the tag “Italy” can be used to categorize *pictures taken in Italy* (in a picture database) or *customers living in Italy* (in a tagged address data base). Ontologies, on the contrary, require a clear and context-independent notion of what it means to be an instance of a respective class.

In this paper, we suggest taking an integrated approach of combining five types of resources and techniques for improving the construction of domain ontologies. We propose to exploit (1) the statistical analysis of folksonomies and the wealth of data resulting from their construction, usage, and the underlying social relationships between actors by providing a set of tools and techniques that identify structural patterns in folksonomies, (2) on-line lexical resources like dictionaries, Wordnet, Google, and Wikipedia; (3) ontologies and Semantic Web resources, (4) ontology mapping and matching approaches, and (5) functionality that helps the community in achieving and maintaining consensus.

The structure of the paper is as follows. In section 2, we give an overview of potential resources and techniques that are available for lifting folksonomies to the level of ontologies. In section 3, we explain the FolksOntology approach that is based on the integration of these elements and the involvement of the community. In section 4, we give a preliminary assessment of the possible contribution of each resource and technique. In section 5, we discuss our proposal in the light of related work, identify future research challenges, and summarize the main findings.

2. Resources for Lifting Folksonomies to the Level of Ontologies

In this section, we give an overview of promising resources that can be exploited for deriving ontologies from folksonomies. There exist at least three groups of such resources: First, folksonomies and their associated data (subsection 2.1); second, online lexical resources (subsection 2.2); and third, ontologies and other Semantic Web resources (subsection 2.3). In subsection 2.4, we discuss how mapping and matching techniques can support the process.

2.1. Folksonomies and Associated Data

Quite clearly, tagging generates more data than merely tags. When we look at Web sites that have an inherent tagging feature, we can see that there are four groups of entities involved in the tagging process: (1) tags, (2) objects, like images or bibliographic references, (3) actors, and (4) the folksonomy-driven Web sites or systems[1] themselves [5]. There is interaction between those entities, which generates a large amount of potentially valuable data, as described in the subsections below.

2.1.1. Folksonomies and Social Networks in One System

During the tagging process, actors are assigning tags to objects (figure 1). The actors describe an object using their own, freely chosen keywords, usually in order to facilitate a later retrieval process. As a consequence, the tags are expressing and reflecting the actors' subjective level of knowledge on and their interest in the respective object.

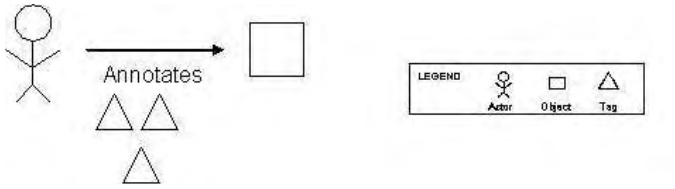


Fig. 1. The Tagging Process

In the past few years, there have been successful attempts of enriching tags with hierarchical relations [6] and the creation of faceted ontologies [7] through studying the use of objects and tags in a system. However, more information is available than merely tags, as explained e.g. in [8], in which the social dimension of actors was introduced. Out of a tripartite model of tags, objects, and actors, three bipartite graphs were generated based on the co-occurrence of its elements: the AC (actor-tag) graph, AI (actor-object) graph, and the CI (tag-object) graph. The folding of these graphs into one-mode networks generates implicit social networks, a network of instances and lightweight ontologies. [8] examines these two lightweight ontologies (one based on sub-communities of interest and another on object overlaps) on a data set of the deli.cio.us system and reveals broader/narrower relations. The authors concluded that analyzing a lightweight ontology of a sub-community is a good mean for discovering

[1] In the rest of the paper, we will use the term systems.

the emergent semantics of a community. Therefore, consolidating and analyzing the user-created data of sub-communities seems a valuable start data set for the creation of ontologies of this sub-group.

We argue that the implicit social networks in a system, which are not studied in [8], may return additional significant information. In particular, one can safely assume that actors are indirectly linked with others by sharing the same tags and/or objects. For example, as shown in figure 2, actors A and B are linked by tag3 and actors B and C are related because they both have tagged object5. In the first case, the social binding is the common language, in the second case, it is the interest in the same objects.

Analyzing such data might reveal relevant relations that can help us in reconstructing an ontology for the respective domain of interest. For instance, there might be a significant relation between object1 (annotated by actor A) and object5 (annotated by actor B): and maybe the tags should be consolidated. Furthermore, a relation might exist between tag3 and the tag set (tag4, tag5, tag6) since they are all used to annotate object5.

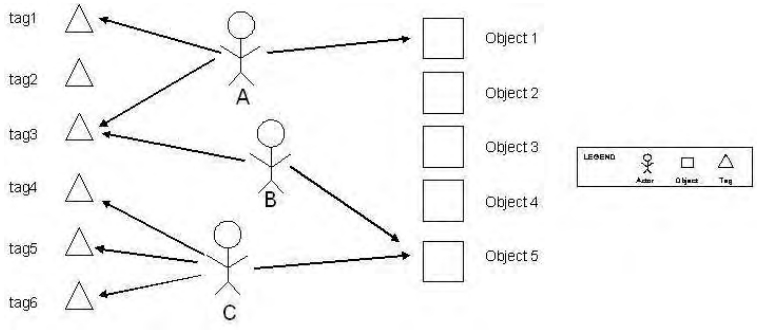


Fig. 2. The Collective Tagging Process

Sometimes, actors have already made explicit their area of interest or expertise, e.g. by joining one or more user groups on the system, which is a feature in some systems (e.g. Bibsonomy, Flickr, YouTube). By that, actors with similar interests can share their objects and tags. However, since everyone may create a new group, reduntant groups and a topic overlap between groups is likely. On Flickr, many groups are discussing and generating tags on similar kind of subjects - there exist, e.g., more than 1290 public groups on wine[2]. Therefore, aggregating the data from those groups may reveal valuable data for the creation of wine ontologies.

Actors can also make their relations and interests public by inviting other actors to their network, as is supported e.g. by deli.cio.us. Adding an actor to your network implies you are having the same interests as this actor, or that there exist some other social bonds. When all the actors are making their interests public, more information can be extracted.

[2] http://www.flickr.com/search/groups/?q=wines retrieved on April 1, 2007.

2.1.2. Folksonomies and Social Networks in Several Systems

As already mentioned, there is a fourth type of entities involved in the tagging process, i.e., *systems*. Since more and more systems are emerging, we believe that tagging data on similar topics and objects is created in parallel on different systems.

Systems are *implicitly* connected through shared sub-communities of interest or common objects. Sub-communities are not exclusively related to just one system. For instance, a sub-community on wines may exist on Flickr as on deli.cio.us. However, we have to be careful when comparing data from different kinds of systems, since a folksonomy can be broad or narrow [3]. In case the actor and creator are both the same, as is the case on Flickr, the consolidated tags constitute a narrow folksonomy. On deli.cio.us every object is tagged by, depending on the popularity of the object, several actors and the aggregation of the tags lead to a broad folksonomy. On the other hand, there may exist implicit links between systems because the actors are annotating the same sets (or kinds) of objects. For instance, the same scholarly publications are tagged on different systems (e.g. Bibsonomy and CiteULike). Consolidating the entire user-created data of similar kinds of objects, which is dispersed on several systems, may generate a more complete overview on the meta data of overlapping objects.

On the other hand, some systems are also *explicitly* connected through explicit social networks of their actors. Information on a person can be given e.g. using FOAF. FOAF allows everyone to describe him/herself (e.g. name, family name, friends), online accounts, groups and documents in a lightweight formal way[3]. Extracting the information that is stored in FOAF profiles can unveil the explicit social networks. The explicit social networks can be used for determining people with shared objects and tags. In [18] a system is proposed where actors can next to tagging their bookmarks, explicitly describe their relations with other people by FOAF. Then, they can import the tags of their friends and establish mappings between their tags and those of their peers. Doing this implies a certain level of trust and can enhance the feedback functionality in the bookmark system. In that way, [18] are trying to create a community-based ontology that is based on explicitly described relations and trust.

We can conclude that this tagging process produces several kinds of data sets that can be analyzed to exploit the information hidden in these systems. It is obvious that the design of proper tools for exploiting structural patterns in folksonomies is a core challenge for tapping this potential.

2.2. Online Lexical Resources

The data sets obtained from the previous resource can be complemented with information from lexical or terminological resources such as Leo Dictionary, Wordnet, Google, and Wikipedia.

Dictionaries are generally considered as a valuable and reliable resource containing definitions of several common words. Nowadays, several dictionaries are online accessible such as Leo Dictionary and the lexical database Wordnet. However, it is not sufficient to rely solely on these resources. For example, rather new or very specific words such as *folksonomy* can not be retrieved although the latter is an established term on the Web. Thus, we should exploit other lexical resources the Web

[3] http://xmlns.com/foaf/0.1/#sec-foafvocab retrieved on April 1, 2007.

is offering, e.g. *Google* and *Wikipedia.* Google is providing some kind of dictionary functions. Each time the user is entering a search key word, Google tries to find similar key words [15]. The search results for both queries are compared (the original one entered by the user and the similar ones). In case the alternative spelling has more hits, a suggestion is made to the user. For instance when typing in the query *occurence*, Google will make the suggestion *occurrence* since the number of results for the user key word *occurence* are significant lower. This suggestion feature is based on the principle of collective wisdom: if the majority of the Web community is using this key word, it is accepted as an existing and well-spelled word. The principle of collected wisdom can also be used for checking the proper usage of language, e.g. for finding proper prepositions. It can be futher improved by considering the region of origin and the authority of the returned Web pages (the page http://www.bbc.co.uk will have a higher credibility than on http://yahoo.com/users/pmiller.htm). The Google dictionary function can be complemented with Wikipedia, the online collaborative encyclopedia, for the identification of words. Everyone can edit and make a new Web page in this user-created encyclopedia. For instance, for "folksonomy", a Wikipedia article was already created in November 2004, whereas the respective word does still not exist in regular dictionaries[4]. With more than 5,300,000 articles [9] in various languages, Wikipedia constitutes a huge corpus of knowledge. In the English language, 1,710,088[5] articles can be identified by a URI; plus it has been shown in [2] that the conceptual meaning of the articles does not change in most cases and thus Wikipedia URIs can be regarded as authoritative identifiers for many concepts.

2.3. Ontologies and Semantic Web Resources

After consulting all the lexical resources, ontologies and Semantic Web resources can be employed as the second level of resources. Freely available ontologies can be retrieved e.g. through the Semantic Web search engine Swoogle. This search engine is searching and indexing Semantic Web documents written in RDF and OWL. It indexes the metadata of the documents and computes relationships between them [10].

Wordnet, which we mentioned in the previous section, can also be exploited as a freely available thesaurus, for which an OWL transcript is available[6]. Wordnet provides an overview of terms and their relationships (e.g. synonyms, meronyms and homonyms). It is often suggested and applied in research papers for extracting semantic information (e.g. in [11], Wordnet is employed for finding synonyms and related terms in order to reduce the communication obstruction between intelligent agents with different ontologies, and [12] use Wordnet to add a conceptual meaning to the tags when annotating a bookmark) .

2.4. Ontology Mapping and Matching Approaches

Next to resources, we can build on established techniques for ontology matching and mapping. In principle, matching of conceptual elements in two ontologies can be

[4] Merriam Webster Online, Leo Dictionaries
[5] http://en.wikipedia.org, retrieved on March 27, 2007
[6] http://www.w3.org/TR/wordnet-rdf/, retrieved May 9, 2007

based either on the labels or on the ontology structure, or both. For deriving ontologies from folksonomies, those techniques may be used in particular for identifying relationships between tags, between tags and lexical resources, and between tags and elements in existing ontologies. [13] describe the theory of formal classification, where labels are translated to a propositional concept language. Each node is associated to a normal form formula that describes the content of the node. This approach is able to capture knowledge that exists implicitly within simple classification hierarchies. [14] describe semantic matching, an approach to matching classification hierarchies. This approach is focused to the graph representation of ontologies, which means it cannot be directly applied to tag data. [15] present the FCA-Merge method, where the input to the method is a set of documents from which concepts and the ontologies to be merged are extracted using natural language techniques. These documents should be representative of the domain at question and should be related to the ontologies. They also have to cover all concepts from both ontologies as well as separating them well enough.

3. The FolksOntology Approach

In this section, we describe (1) *how* the resources from the previous section can be fully exploited for making ontologies out of folksonomies and (2) *how* the community can be involved as a mechanism to validate all the information extracted from the resources.

3.1. Fully Exploiting the Resources

A first principle of our approach is that we try to integrate every reasonable data resource and invokable functionality from the Web that can help us construct ontologies from the social interaction taking place on the Web. In other words, we want to take the vast amount of evidence created by users contributing to the Web and extract consensual conceptualizations from that.

3.1.1. Cleansing and Preparation of Tags

Before analyzing all the data sets of folksonomies, we must clean tag sets. Since actors can choose any keyword for categorizing their content, they are applying their own spelling and tagging rules (e.g. singular or plural nouns, conjugated verbs). As a consequence, tags are polluted and need to be cleansed. This can be performed through stemming algorithms. These algorithms are reducing tags to their stem or root. It is important not to loose the context of the tags, therefore the stemming process of tags should be limited to plural nouns and conjugated verbs. After this stemming algorithm, it has to be checked whether all the tags are spelled correctly. We can use the four lexical resources Leo Dictionary, Wordnet, Google, and Wikipedia to check whether or not the tags are misspelled. In case a tag is not retrieved in any of these resources, the frequency of this tag should be counted. A low frequency may indicate that the tag is misspelled and a high frequency can be an indication of the offset of a new word created in the tagging community. This word should be added to the list of new words that has to be examined by the community (subsection 3.2).

3.1.2. Statistical Analysis of Folksonomies, Usage Data, and Social Networks

In this paragraph we give an overview of data sets described in section 2.1 and explain the objective, input, output, and techniques that can be employed.

Table 1. Statistical analysis of tagging data on a single system

Step	Objective	Input	Output	Techniques
1	Determining pairs of tags	Tags, tag/object data	Pairs of tags	Co-occurrence technique: each time two tags are used to tag the same object, the tie strength between two tags is increased [19].
2	Enriching tags	Objects and Tags	a) Hierarchical relations between tags b) faceted ontology	a) [7] presents an algorithm based on the cosine similarities between tags. Tags are aggregated in tag vectors and the cosine similarity calculates the angle between two tag vectors. The smaller the angle, the more similar the tags are. The tags are consequently placed as a node in a similarity graph. If the similarity of two tags exceeds a threshold value, the two nodes are connected with an edge. A hierarchical taxonomy can be deducted from the similarity graph. b) A combination of co-occurrence between tags and a subsumption-based model is presented in [6].
3	Analyzing and creating sub-communities	Actors and tags	Lightweight ontologies based on community overlap	1) [8] folds the AC Graph (actor tags Graph) into a network based on tags. The weights of tags are calculated by the number of times the actors have used the tags in combination. [8] uses social network analysis measures (such as degree, closeness and betweenness centrality) to determine the general and specialized tags. General tags are used to bridge two clusters and specialized tags are parts of a specific cluster. Clustering techniques are used to determine the synonyms of the specialized tags. [8] uses set theory to determine the broader/narrow relations in the subcommunity
4	Analyzing social networks based on shared objects	Actors and objects	Clusters of actors with shared objects	1) Analyzing a social network. The tie strength between actors is measured by the number of times the actors have tagged the same object. Social network measures and/or clustering techniques can be used for determining the clusters of actors with similar tagged objects. 2) Analyzing the objects of the actors in each cluster: text mining techniques, digital photo similarity analysis
5	Analyzing social networks based on shared tags	Actors, tags, and objects	Clusters of actors with shared tags	1) Analyzing a social network. The tie strength between actors is measured on the number of times the actors have used the same tag. Social network measures and/or clustering techniques can be used for determining the clusters of actors using the same tags. 2) All the tags used by the actors of a cluster can be further analyzed by using the technique described in step 1
6	Merging similar	Groups (+tags,	Clusters of similar groups	1) The groups can be clustered by setting up a network analysis with groups instead of actors.

	groups	objects, actors)		However, the analysis has to be performed on data sets of equal size. This means if the size of the different groups (=number of tags) are differring, the frequency of tags has to be adjusted in proportion. The tie strength between two groups is calculated on the basis of shared tags. Social network measures and/or clustering techniques can be used for determining the clusters. 2) These clusters can be further analyzed by using the technique described in data set 1
7	Analyzing explicit social network	Actors and their relations	Clusters of actors	1) Analyzing the social network. The tie strength between actors can be 0, 1 or 2 depending on the fact of two persons have linked to each other. 2) These clusters can be further analyzed by using the technique described in step 1

Table 2. Statistical analysis of tagging data across multiple systems

Step	Objective	Input	Output	Method
1	Analyzing and creating sub-communities	Actors and tags of different systems	Clusters of communities with similar interests	1) The same techniques as described above can be employed. However, the analysis has to be performed on data sets of equal size. This means if the tags "size" of the different systems are differing, the frequency of tags has to be adjusted in proportion. 2) These clusters can be further analyzed by using one of the techniques described in step 1 in Table 1.
2	Analyzing communities of shared objects	Actors and objects of systems with the same annotated objects	Clusters of communities on overlapping objects	1) The same techniques as described above can be employed, except that the weights of the objects are calculated by the number of times the actors have used the objects in combination. However, the analysis has to be performed on data sets of equal size. This means that if the size of the different systems is differring, the proportions have to be adjusted. 2) These clusters can be further analyzed by using the technique described in step 1 in Table 1.
3	Analyzing the explicit social network	Actors (FOAF)	Clusters of actors	We can take the direct RDF data for determining social proximity.

3.1.3. Exploiting Online Lexical Resources

The tag data set obtained from the previous steps can be enriched by using the online lexical resources as described in section 2.2. However, these lexical resources can also be used for other purposes than merely spelling checks (except for Google). Tags can be replaced by concepts and homonyms, or translated from a foreign language into English as is elaborated in the following paragraphs.

Wikipedia: Wikipedia articles are identified by URIs which can be regarded as reliable identifiers for conceptual entities [2]. The meaning of those entities is

described in natural language and augmented by multimedia elements and agreed upon by a large community. Hence, Wikipedia is the biggest available collection of conceptual entities that are described with natural language and identified by URIs. Already having unique identifiers (e.g. URIs) assigned to concepts defined only in natural language is very beneficial, for it helps improve recall and precision in information retrieval by avoiding synonyms and homonyms. Additionally, Wikipedia contains disambiguation pages in order to deal with homonyms. When one word has several meanings, the meanings are collected on a disambiguation page in order to lists articles associated with the same title. This feature can be used to identify and deal with homonyms. Wikipedia also contains an implicit and evolving multilingual dictionary, since a Wikipedia page can have links that refer to the same topic in another language. These links can be retrieved in an XML format easily with the Wikipedia export function[7].

Leo dictionaries: Leo (Link everything online) provides a translation service for German, English, French, and Spanish. This functionality can be used for dealing with different languages. Additionally, Leo contains a definition of terms in German.

Wordnet can be used to deal with synonyms and homonyms: words with similar or identical meaning must be mapped to each other (e.g. baby and infant). Furthermore, words that have different conceptual meanings (e.g. Jaguar as the car and the animal) can be identified with Wordnet as well.

3.1.4. Ontologies and Semantic Web Resources

The tag sets obtained in subsection 3.1.2 can also be enriched by trying to establish mappings to elements in existing ontologies. Also, the explicit relationships in existing ontologies may be reused, e.g. for determining whether a hierarchical relation holds between two terms. In particular, the Swoogle engine can be used to query for ontologies and ontology usage data.

3.1.5. Mapping and Matching approaches

The formal classification theory of [13] can be employed for mapping the labels of existing classifications with the tags obtained from the folksonomies. Consequently, we can also use the lexical resource Wordnet to create a mapping with an existing ontology.

3.2. Mechanisms for Involving the Community

Instead of aiming at the fully automated creation of ontologies from folksonomies, we suggest a semi-automated approach, in which the aforementioned techniques are combined with collective human intelligence. In other words, we propose that (1) the results from the previous stages have to be confirmed by the community and (2) information that could not be retrieved from the resources (e.g. relations between tags) may be contributed by the community on demand. For this, we can combine visualization techniques and implicit and explicit voting mechanisms on conceptual choices. For example, a concept hierarchy reconstructed from data could be presented

[7] http://de.wikipedia.org/wiki/Spezial:Exportieren retrieved on April, 1 2007.

to the users on a separate Web page, but respective subClassOf relations would only be created if the community approves this.

4. Overview of the Contribution of Each Resource and Technique

In this section, we give a preliminary evaluation of the potential contribution of the various resources and techniques. In Table 3, we summarize the type of contribution that available techniques can provide. In Table 4, we assess the size of lexical and structural data sources that we propose to exploit. While the mere size of a resource is not always and advantage, we assume that in here, a large size makes a resource more attractive for our approach.

Table 3. Type of contribution of each technique

Technique	**Type of Contribution**
Ontology matching algorithms	Finding equivalences between labels or between conceptual elements in graphs
Co-occurrence technique	Finding tag pairs
Co-occurrence technique + Subsumption model	Creating a faceted ontology of tags
Social Network Analysis techniques + set theory	Lightweight ontologies based on community overlap
Social network techniques	Creating a) Clusters of actors with shared objects b) Clusters of actors with shared tags c) Clusters of similar groups d) Clusters of actors that have explicitly indicated their relationship
Visualizations	Visualization of ontologies helps user to grasp the intention of concepts.
Discussion and voting	Like on Wikipedia, users can remove disputes by performing discussions and then vote on the result.

Table 4. Type of contribution and size of available resources

Resource	Type of Contribution	Size
Wikipedia entries in multiple languages	Since Wikipedia contains a wealth of mutual links between pages in multiple languages that cover the same topic, we can exploit this for the unique identification of conceptual entities and for spelling checks.	5,300,000 [9] entries in total 1,710,088[8] English articles
Wikipedia disambiguation pages	Indicators for homonyms	6,67% of English articles [2]
Overlap of multiple folksonomy-driven websites targeting at the same type of objects	Finding similar tagged objects, e.g. tags referring to the same scholarly publication.	No information available
Tags	Raw set of candidate concepts	We were unable to get information on the total amount of deli.cio.us and Flickr tags – for Technorati, we at least know that there exist more than 81 Million posts[9].
Annotations	Finding tag-object patterns	Delicious: 53.000.000[10] Flickr: No data available Technorati: : 27 million weblogs[11]
Actors	Finding users with similar interests and vocabulary	Delicious: 90.000[12] Flickr: No data available Technorati: also about 27 million (if we assume that every actor has, on average, only one weblog)
Google suggestions	Spell checks	No information available
WordNet	Mapping synonyms, retrieving descriptions of terms, ancestors	27 semantic properties[13]
Swoogle	Finding related ontologies and annotations	More than 10,000 ontologies[14], though many of questionable maturity
Leo Dictionaries	Translation of terms	453,994 entries[15]

[8] http://en.wikipedia.org, retrieved on March 27, 2007
[9] http://technorati.com/weblog/2006/02/81.html , retrieved on April 3, 2007.
[10] http://www.techcrunch.com/2006/08/04/more-stats-on-delicious-this-time-positive/#comments, retrieved on April 2, 2007.
[11] http://technorati.com/weblog/2006/02/81.html , retrieved on April 3, 2007.
[12] http://www.pui.ch/phred/archives/2005/05/delicious-statistics-that-is-extrapolation.html, retrieved April 3, 2007.
[13] http://www.w3.org/TR/2006/WD-wordnet-rdf-20060619/#details, retrieved on April 2, 2007.
[14] http://swoogle.umbc.edu/, retrieved on March 29, 2007.
[15] http://dict.leo.org, retrieved on March 29, 2007

Of these resources, Google, Wikipedia, and Wordnet can be accessed either by APIs or straightforward screen-scraping techniques. For Leo, there is currently no API access supported.

5. Discussion and Conclusion

In this section, we compare our proposal to previous works, evaluate the added value, and identify future research steps.

Our work is closely related to [15]. In [15] the authors are presenting an approach to enrich tags with semantics in order to integrate folksonomies and the Semantic Web. Similarly to our approach, they are also using online lexical resources, ontologies and Semantic Web resources for amending the tags. Our approach extends this direction, since, first, we suggest deriving actual ontologies out of folksonomies, while [15] focuses on using existing resources and ontologies to map tags into concepts, properties or instances and determine the relations between these mapped tags. Second, we suggest to consider the varying resources not only as an isolated source helping in a single step of the tag processing but to channel all the social interaction manifested on the Web in such resources as the main input for automatically creating and maintaining domain ontologies. Third, we suggest to continuously involve human intelligence in the form of community approval of the resulting conceptualization, in order to confirm the semantics obtained from existing ontologies and resources.

Putting the community in the center of the ontology engineering process has already been proposed in [16] and [17], and other work on collaborative ontology engineering. In [16] and [17], the authors are generating a community-driven ontology based an ontology maturing process. This process contains the following steps: 1) community members are generating new ideas and related terminology through the tagging process, 2) the new tags and their concept definitions are presented and discussed in the whole community: everyone can change a definition, add synonyms etc., 3) the textual concept definitions created in the second phase, are formalized and hierarchical relations are added. In [17], axiomatization as a fourth phase is added to the process. During this step, additional semantics are added. In [17] two tools are presented that are based on this ontology maturing process. One of the discussed tools is using visualizations and another is using wiki technology to support the formation of consensus in the community. This approach differs from ours since they are not relying on existing resources for reuse. They are generating ontologies from scratch. In a nutshell, our approach aims at combining the strengths of [16, 17] and [15] and fully using a “mash-up” of available lexical, semantic, and social data sources for producing and maintaining domain ontologies

Acknowledgements: The work presented in this paper has been supported by the European Commission under the projects SUPER (FP6-026850) and MUSING (FP6-027097), and by the Austrian BMVIT/FFG under the FIT-IT project myOntology (Grant no. 812515/9284). Martin Hepp is also supported by a Young Researcher’s Grant (Nachwuchsförderung 2005-2006) from the Leopold-Franzens-Universität Innsbruck.

References

[1] Hepp, M.: *Possible Ontologies: How Reality Constrains the Development of Relevant Ontologies*. IEEE Internet Computing, 2007. **11**(7): pp. 96-102.

[2] Hepp, M., D. Bachlechner, and K. Siorpaes: *Harvesting Wiki Consensus - Using Wikipedia Entries as Ontology Elements*, in: *Proceedings of the Workshop on Semantic Wikis at the ESWC2006 (ESWC2006)*, Budva, Montenegro, 2006.

[3] Vander Wal, T.: *Folksonomy*. Available from http://vanderwal.net/folksonomy.html, retrieved March 30, 2007.

[4] Golder, S. and B.A.Huberman: *Usage Patterns of Collaborative Tagging Systems*. Journal of Information Science, 2006. **32**(2): pp. 198–208.

[5] Gruber, T.: *Folksonomy of Ontology: A Mash-up of Apples and Oranges*. *First On-Line conference on Metadata and Semantics Research (MTSR2005).* Available from: http://tomgruber.org/writing/mtsr05-ontology-of-folksonomy.htm, retrieved March 30, 2007.

[6] Schmitz, P.: *Inducing Ontology from Flickr Tags*, in: Proceedings of the *Collaborative Web Tagging Workshop at the 15th WWW Conference (WWW2006)*, Edinburgh, Scotland, 2006.

[7] Heymann, P. and Hector Garcia-Molina: *Collaborative Creation of Communal Hierarchical Taxonomies in Social Tagging Systems*. Stanford InfoLab Technical Report 2006-10.

[8] Mika, P.: *Ontologies Are Us: A Unified Model of Social Networks and Semantics*, in: *Proceedings of the 4th International Semantic Web Conference (ISWC2005).* LNCS 3729, Springer-Verlag, 2005.

[9] Wikipedia Foundation: *About Wikipedia*. Available from: http://en.wikipedia.org/wiki/Wikipedia:About#Contributing_to_Wikipedia, retrieved March 30, 2007.

[10] Ding, L., et al.: *Swoogle: A Search and Meta Data Engine for the Semantic Web*, in: *Proceedings of the Thirteenth ACM Conference on Information and Knowledge Management (CIKM'04)*, Washington D.C., USA, 2004.

[11] Bailin S. and W. Truszkowski: *Ontology Negotiation between Agents Supporting Intelligent Information Management*, in: *Proceedings of the Workshop on Ontologies in Agent-based Systems* at the Fifth International Conference on Autonomous Agents (Agents 2001). Montreal, Canada,2001.

[12] Spyns, S., et al.: *From Folksologies to Ontologies: How the Twain Meet*, in: *On the Move to Meaningful Internet Systems 2006: CoopIS, DOA and ODBASE (OTM2006)*, Montpellier, France: Springer, 2006.

[13] Giunchiglia, F., M. Marchese, and I. Zaihrayeu: *Towards a Theory of Formal Classification*, in: *Proceedings of the AAAI-05 Workshop on Contexts and Ontologies: Theory, Practice and Applications (C&O-2005)*, Pittsburgh, Pennsylvania, USA, 2005.

[14] Giunchiglia, F. and P. Shvaiko: *Semantic Matching.* The Knowledge Engineering Review, 2004. **18**(3): pp. 265-280.

[15] Stumme, G. and A. Maedche: *Ontology Merging for Federated Ontologies on the Semantic Web*, in: *Proceedings of the International Workshop for Foundations of Models for Information Integration (FMII2001)*, Viterbo, Italy, 2001.

[15] Specia, L. and E. Motta: *Integrating Folksonomies with the Semantic Web*, in: *Proceedings of the European Semantic Web Conference (ESWC 2007)*, Innsbruck, Austria: Springer, 2007.

[16] Maier, R and Schmidt, A.: *Characterizing Knowledge Maturing: A Conceptual Process Model for Integrating E-Learning and Knowledge Management***,** in: *Proceedings of the 4th Conference Professional Knowledge Management - Experiences and Visions (WM '07)*, Potsdam, Germany, 2007.

[17] Braun, S., et al.: *Ontology Maturing: A Collaborative Web 2.0 Approach to Ontology Engineering*, in: *Proceedings of the Workshop on Social and Collaborative Construction of Structured Knowledge (CKC) at the 16th International World Wide Web Conference (WWW 2007)*, Banff, Alberta, Canada, 2007.

[18] I. Ohmukai, Hamasaki, M., and Takeda, H. *A Proposal of Community-based Folksonomy with RDF Metadata*, in: *Proceedings of the Workshop on End User Semantic Web Interaction*, co-located with the Fourth International Semantic Web Conference (ISWC2005), Galway, Ireland, 2005.

[19] T. Coenen, et al.: *Knowledge Sharing over Social Networking Systems: Architecture, Usage Patterns and their Application*, in: *Proceedings of the OTM Workshops 2006 (OTM2006)*, LNCS 4277, pp. 189–198, Berlin, Heidelberg, Springer-Verlag, 2006.

Folksonomies, the Semantic Web, and Movie Recommendation

Martin Szomszor[1], Ciro Cattuto[3,2], Harith Alani[1], Kieron O'Hara[1], Andrea Baldassarri[2], Vittorio Loreto[2,3], Vito D.P. Servedio[2,3]

[1]School of Electronics and Computer Science
University of Southampton, SO17 1BJ, UK
[2]Dipartimento di Fisica, Università di Roma "La Sapienza"
P.le A. Moro, 2, 00185 Roma, Italy
[3]Museo Storico della Fisica e Centro Studi e Ricerche Enrico Fermi
Compendio Viminale, 00184 Roma, Italy

Abstract. While the Semantic Web has evolved to support the meaningful exchange of heterogeneous data through shared and controlled conceptualisations, Web 2.0 has demonstrated that large-scale community tagging sites can enrich the semantic web with readily accessible and valuable knowledge. In this paper, we investigate the integration of a movies folksonomy with a semantic knowledge base about user-movie rentals. The folksonomy is used to enrich the knowledge base with descriptions and categorisations of movie titles, and user interests and opinions. Using tags harvested from the Internet Movie Database, and movie rating data gathered by Netflix, we perform experiments to investigate the question that folksonomy-generated movie tag-clouds can be used to construct better user profiles that reflect a user's level of interest in different kinds of movies, and therefore, provide a basis for prediction of their rating for a previously unseen movie.

1 Introduction

Recommendation systems have evolved in recent years to support users in the discovery of new items through the construction of profiles that represent their interests, and networks that connect them to other users who share similar tastes. Many of these recommendation strategies rely on the modelling of intrinsic attributes about each item (e.g. the keywords for a document or the genre of a CD) so that the items can be categorised, and the level of interest a user has can be expressed in terms of these attributes. This knowledge is usually gathered over time, by monitoring and logging various user interactions with the system (e.g. buying, browsing, bookmarking). Amazon.com, for example, provides a recommendation service that is based on collaborative filtering: if a user buys an item that has been bought by a number of other users in combination with some other items, then those other items will be recommended by Amazon.com to the user. These recommendations are entirely based on what goes on inside the system (Amazon.com in this case), ignorant of any external knowledge about the items or the users themselves.

To improve on such recommendation techniques, we think it might be useful to incorporate data from as many sources as possible to build richer profiles that model many facets of interest that might be difficult and impractical to capture by a single system or service. In recent years, many Web 2.0 applications, such as folksonomies and blogs, have become popular places where individuals provide and share various type of information. This information may, directly or indirectly, represent the interests of those individual users. There could be much to learn about a user from analysing their shared profile in MySpace, bookmarks in del.iciou.us, photos in Flickr, references in Connotea, and any other popular Web 2.0 applications.

Although folksonomies provide structures that are considered to be formally weak or unmotivated, they do have two advantages in this particular context. First, they are strongly connected with the actual use of the terms in them and the resources they describe. And second, they are relatively cheap to develop and harvest, as they emerge from individual tagging decisions that are cheap for the user. To that extent, they may provide data about the *perceptions* of users, which is what counts in this particular recommendation context. In this respect, a folksonomy, we hypothesise, will be of greater value than an ontology of films, which might provide a more objective sense of whether two films are similar, but which need not map onto viewers' perceptions. However, it is important to note that the Semantic Web and folksonomies are not in competition here; folksonomies are not "cheaper" or "simpler" or "bottom up" versions of ontologies. As the system we will be experimenting with brings together data from a number of sources, then Semantic Web technology is certainly required.

In this paper, we test the hypothesis that folksonomies describing movie titles can be used as a basis for building recommendation profiles by associating each user with *tag-clouds* that represent their interests. By using Semantic Web technology to integrate heterogeneous data sources, a large collection of movie titles, ratings on these titles, and tagging data is assembled, providing the basis for empirical testing of our algorithms.

This paper is organised as follows: Section 2 contains a brief literature survey on tagging systems, the Semantic Web, and recommendation strategies, and is followed by a description of our architecture in Section 3. In Section 4, we present our recommendation algorithm that is based on the construction of interest tag-clouds before the results of our experiments are shown in Section 5. Finally, Section 6 contains our conclusions and directions for future work.

2 Background

2.1 Folksonomies

The term *folksonomy* was first coined by T. Vander Wal [23] to describe the taxonomy-like structures that emerge when large communities of users collectively tag resources. These *folk taxonomies* reflect a communal view of the attributes associated to items, essentially supplying a bottom-up categorisation of resources [14, 10]. Since individuals from different communities utilise different

tags, often reflecting their degree of knowledge in the domain, folksonomies can support highly personalised searching and navigation. For example, an article in the social bookmarking site `del.icio.us` [7] concerning web programming may have the tags `programming`, `ajax`, `javascript`, `tutorial`, and `web2.0`. With tags describing resources at varying levels of granularity, users may seek out their desired resources using terms they are familiar with.

Examination [8] of social tagging sites, such as `del.icio.us`, has revealed a rich variety in the ways in which tags are used, allowing tags themselves to be categorised in a number of ways:

- Tags may be used to identify the topic of a resource using nouns and proper nouns such as `news`, `microsoft`, `vista`.
- To classify the type of resource (e.g. `book`, `blog`, `article`, `review`, `event`).
- To denote the qualities and characteristics of the item (e.g. `funny`, `useful`, and `cool`).
- A subset of tags, such as `mystuff`, `myphotos`, and `myfavourites`, reflect a notion of self reference, often used by individuals to organise their own resources.
- Much like self referencing tags, some tags are used by individuals for task organisation (e.g. `to read`, `job search`, and `to print`).

Another important aspect of tagging systems is how they operate. Marlow *et al* [13] provide an extensive classification of tagging systems that enables us to compare the benefits and deficiencies of different systems according to seven characteristics:

1. **Tagging rights**
   The permission a user has to tag resources can effect the properties of an emergent folksonomy. The spectrum of tagging permissions ranges from *self-tagging*; where users are only allowed to tag the resources they have created, to a tagging *free-for-all*; where users may tag any resources. Some compromise between the two may be supported, for example, by allowing users to tag resources created by those in their social network.
2. **Tagging Support**
   One important aspect of a tagging system is the way in which users assign tags to items. They may assign arbitrary tags without prompting (*blind tagging*), they may add tags while considering those already added to a particular resource (*viewable tagging*), or tags may be proposed (*suggestive tagging*). While it has been shown [8] that suggestive tagging results in faster convergence to a folksonomy, it is not clear whether it effects the quality or diversity.
3. **Aggregation**
   Tagging events may be recorded at different levels of granularity. For example, all tagging events may be uniquely logged, keeping track of all the tags assigned by all of the users (the *bag-model*). This method allows tag weighting to be derived to reflect the popularity of a given tag for a particular resource. On the other hand, a simple *set-model*, resource centric approach

may be used where a set of tags is maintained for each resource, meaning the popularity of assignment for each tag is unknown.

4. **Types of Object**
   The types of resource tagged allow us to distinguish different tagging systems. Popular systems include Web pages (`del.icio.us`), bibliographic data (`CiteULike`), blogs (`technorati`), images (`flickr`), video (`You Tube`), audio objects (`last.fm`), and movies (`imdb`, `movielens`).
5. **Sources of Material**
   Tagging systems may allow users to upload resources (e.g. `You Tube`), or resources may be managed by the system (e.g. `last.fm`, `imdb`). In some situations, such as `del.icio.us`, arbitrary Web resources may also be referenced.
6. **Resource Connectivity**
   Within a tagging system, resources may be connected independently of their tags. For example, Web pages may be connected via hyperlinks, or items may be grouped together (e.g. photo albums in `flickr`). When such linking occurs, additional analysis can reveal correlations between items that correspond with the co-occurrence of tags.
7. **Social Connectivity**
   Finally, it is useful to consider how users of the system may be connected. Many tagging systems include social networking facilities that allow users to connect themselves to each other based on their location, areas of interest, educational institutions and so forth. These social networks provide an excellent opportunity to explore the correlation between localised substructures in folksonomies and social connectivity.

2.2 Semantic Web

The Semantic Web (SW) has proven to be a useful data integration tool, facilitating the meaningful exchange of heterogeneous data, particularly in areas such as e-science and medicine. However, as is well known, there are costly overheads in the use of the SW; in particular, the effort involved in building, and maintaining, useful ontologies and acquiring rich and well structured RDF can be relatively high, a fact often blamed for slowing down the wide adoption of Semantic Web technology [5, 1]. Web 2.0, and the notion of community tagging, is showing promise as an alternative way to quickly and cheaply produce structured semantic models [9] through the study of emergent semantics [22]. It has been argued that harnessing the knowledge embedded in folksonomies can lead to building shallow ontologies that are more receptive to knowledge change over time [16].

Nevertheless, we should not think that Web 2.0 and the SW, tags and RDF, folksonomies and ontologies are competing for the same space [2]. Folksonomies are essentially a development in information retrieval, an interesting variant on the keyword-search theme. This makes them particularly interesting in the context of film recommendations: they help answer the question "how can I find films relevant to the concept in which I am interested." Ontologies are tools for

data integration: they are attempts to regulate part of the world of data, and to facilitate mappings and interactions between data held in disparate formats or locations.

The important question with respect to SW technology and Web 2.0 is not how to manage a trade-off, but rather, how to use them together for the best advantage. Much will depend on the particular context of use, but in the case of film recommendation, a fairly basic architecture suggests itself. The use of Web 2.0 data for the purpose of recommendation makes sense, as this emerges from tagging based on perceptions. Folksonomies, being organic structures that mirror the understanding users have of resources, can provide a better foundation for the expression of user's interests. This idea has been investigated in the context of social bookmarking [19] to build a Web Page recommender system and provided encouraging results.

Nevertheless, the hypothesis with which we are working is whether we can improve the performance of recommender systems by giving the systems access to greater quantities of information, which implies the need to integrate relevant data acquired from heterogeneous sources. This immediately suggests a role for SW technologies. As noted, the issues to be addressed in this part of the architecture include the developing a suitable ontology and acquiring RDF without driving up the cost of development.

2.3 Recommender Systems

Recommender systems are usually used in one of two contexts: (1) to help users locate items of interest they have not previously encountered, (2) to judge the degree of interest a user will have in item they have not rated. With the growing popularity of on-line shopping, E-commerce recommender systems [20] have matured into a fundamental technology to support the dissemination of goods and services. Much research has been undertaken to classify different recommendation strategies [6, 11], but for the purposes of this paper, we divide them broadly into two categories.

Collaborative recommendation is probably the most widely used and extensively studied technique that is founded on one simple premise: if user A is interested in items w, x, and y, and user B is interested in items w, x, y, and z, then it is likely that user A will also be interested in item z. In a collaborative recommender system, the ratings a user assigns to items is used to measure their commonality with other users who have also rated the same items The degree of interest for an unseen item can be deduced for a particular user by examining the ratings of their neighbours. It has been recognised that users interest may change over time, so time-based discounting methods have been developed [3, 21] to reflect changing interests.

Content-based recommendation represents the culmination of efforts by the information retrieval and knowledge representation communities. A set of attributes for the items in the system is conceived, such as the keywords and term frequencies for documents in a repository, so the system can build a profile for each user based on the attributes present in the items that user has rated highly.

The interest a user will have in an unrated item can then be deduced by calculating its similarity to their profile based on the attributes assigned to the item.

Such systems are not without their deficiencies, the most prominent of which arise when new items and new users are added to the system - commonly referred to as the *ramp-up* problem [12]. Since both content-based and collaborative recommender systems rely on ratings to build a user's profile of interest, new users with no ratings have neutral profiles. When new items are added to a collaborative recommender system, they will not be recommended until some users have rated them. Collaborative systems also depend on the overlap in ratings across users and perform badly when ratings are sparse (i.e. few users have rated the same items) because it is hard to find similar neighbours.

Hybrid recommender systems, i.e. those which make use of collaborative and content based approaches, have been developed to overcome some of these problems. For example, collaborative recommender systems do not perform well with respect to items that have not been rated, but content-based methods can be used to understand their relationship to other items. Hence, a mixture of the two approaches can be used to provide more robust systems. More recent recommender systems have also investigated the use of ontologies to represent user profiles [15]. Benefits of this approach are more intuitive profile visualisation and the discovery of interests through inferencing mechanisms.

3 Recommendation Architecture

To gather the information necessary to construct profiles that describe the kinds of movies a user is interested in, we combine data harvested from two sources, and also combine the use of Web 2.0 and SW technology. This section first presents the Web 2.0 data sources we use to construct a knowledge base about movies and how users rate movies (Section 3.1), and second the semantic technologies to represent the information in this knowledge base (Section 3.2).

3.1 Data Sources

For movie tagging data, we make use of the Internet Movie Database (IMDB) [25]; an online database containing extensive information on movies, actors, television shows, and production personnel. IMDB holds information on approximately 960,000 titles and 2,300,000 people, and is the largest known accumulation of data about films [24]. In terms of tagging, IMDB allows users to add *keywords* to titles to describe arbitrary features of the movie. Typically, these are used to denote important scenes in the film (e.g. `sword-fight`, `kidnapping`, `car-chase`), plot themes (e.g. `love`, `revenge`, `time-travel`), locations (e.g. `space`, `california`), film genres (e.g. `independent-film`, `non-fiction`, `cult-favorite`), and background data (e.g. `based-on-novel`, `based-on-true-story`). On average, a popular movie has between 50 and 150 keywords attached to it.

Currently, IMDB uses this tagging data to create a movie search tool that helps users to find popular movies based on their keywords. A screen shot of this interface in shown in Figure 1 and contains two panels: on the left, a tag cloud is

used to display keywords; and on the right, a list of the top movies that contain the currently viewed keywords. In this particular example, the keywords `space` and `android` are used as the search terms.

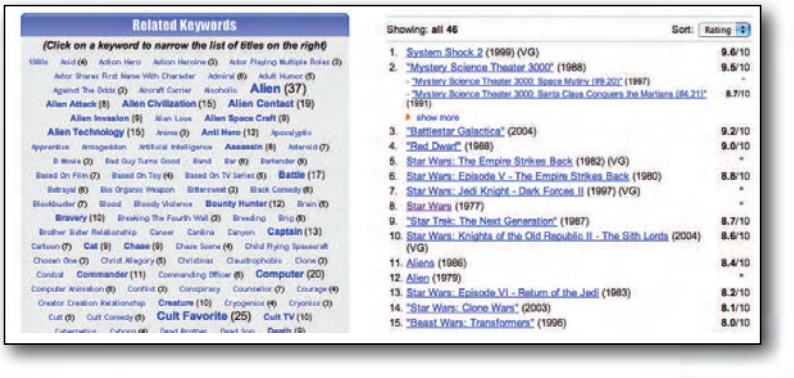


Fig. 1. A screen shot of the IMDB keyword search interface.

With respect to the tagging system categorisation presented earlier in Section 2.1, IMDB is a tagging *free-for-all*. Although the addition of keywords to a movie is moderated, it is used mainly to prevent spam attacks and not to manage the keywords used. When adding keywords to a movie, users can see the keywords that have already been added, but they are not prompted with suggestions (*viewable* tagging support). In terms of aggregation, IMDB falls into the set-model category because the individual keyword assignments by each user cannot be seen. Instead, a simple list of keywords is maintained for each movie and duplicates are not allowed.

To test our keyword-based recommendation approach, we use data provided by Netflix [17] as part of the Netflix Prize [18]. Netflix is an online DVD rental service, established in 1998, the provides a flat rate, mail-based, rental service to customers in the United States. Their current DVD collection contains around 75,000 titles, offered to a customer base of over 6 million individuals. After renting a movie, customers may enter their rating of the movie into the Netflix database via the website, using a discrete score from 1 to 5.

In October 2006, Netflix began a competition to find better recommendation systems, offering a grand prize of $1 million to anyone managing to improve on their own algorithm by 10%. To drive this competition, Netflix published a large set of movie rating data from their database featuring 480,189 customers and 100,480,507 ratings across 17,770 movie titles.

3.2 Data Representation

To combine the IMDB database and the Netflix rating data, we import both data sets into a standard relational database. String matching is then used to correlate the movie titles in the Netflix data dump with their counterparts in the IMDB data set, providing a way to retrieve IMDB keywords for each Netflix movie

title. To provide a homogeneous view over both data sources, an ontology is used in conjunction with the D2RQ [4] mapping technology, supplying a SPARQL end-point which can be queried to find extensive amounts of information on movies such as: the keywords assigned; the actors appearing in the film; the writers, directors and production crew; as well as rating information for movies featured in the Netflix data set. The two perceived issues with semantically-enabled technologies mentioned in 2.2 are thereby addresses. Instead of having to convert data to RDF triples, D2RQ allows this to be done on the fly. Within the well-structured domain of the system, the ontology was deliberately kept as lightweight as possible.

The ontology used is illustrated in Figure 2 where classes depicting IMDB data are shown in white boxes, and classes describing Netflix data are shown in grey boxes. The IMDB data set is centered around the concepts of *Movie*, *Person*, and *Role*. The movie class has properties describing the certificate information, keywords, rating data, and release date information. A Person is anyone who is associated with a movie, i.e. an actor or director, and a Role is used to define how a person is connected to a movie. This abstraction of roles allows the same person to have different functions for the same movie, for example, being a writer and director.

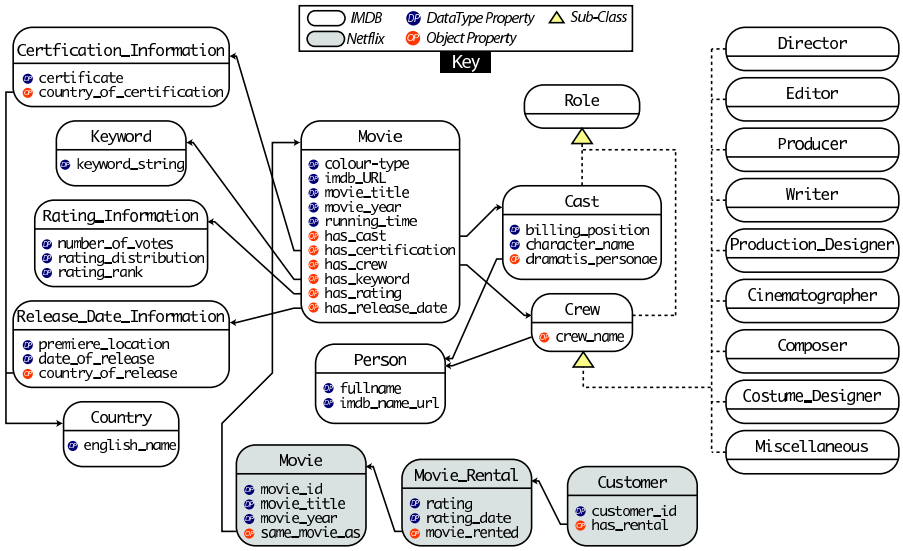


Fig. 2. The ontology used to integrate IMDB and Netflix data.

4 Recommendation Method

To explore the relationship between the way a user rates movies and the keywords that are assigned to movies, we have devised two prediction algorithms that guess

the rating a user would give to a previously unrated movie based on tag-clouds that depict their interests. For comparison, we also specify a naive average-rating algorithm were the average rating for a movie across all users is used as the predicted rating.

4.1 Notation

Let us denote a given user by $u \in U$, where U is the set of all users, a movie by $m \in M$, where M is the set of all available movies, and a rating value by the integer $r \in \{1, 2, 3, 4, 5\} \equiv R$. We indicate the set of movies rated by user u as M_u. On this set we define the rating function for user u as $f_u : m \in M_u \mapsto f_u(m) \in R$.

When keywords or tags are available for a movie m, we denote by K the global set of keywords, by K_m the set of keywords (or tags) associated with movie m, and by N_k the global frequency of occurrence of keyword k for all movies. We can then introduce a notion of *rating tag-cloud* $T_{u,r}$ for a given user u and rating r as the set of couples (k, n_k), where $k \in K$ indicates a keyword (or tag) and $n_k = n_k(u, r)$ is its frequency of occurrence for all movies that user u has associated with rating r. That is,

$$n_k(u, r) = |\{m \in M_u \,|\, k \in K_m \wedge f_u(m) = r\}| \,. \tag{1}$$

Two sample rating tag-clouds are shown in Figure 3; the left one is a rating 1 tag-cloud, and the right one is a rating 5 tag-cloud. The size of keywords is proportional to the logarithm of their frequency of occurrence in the tag-cloud they belong to.

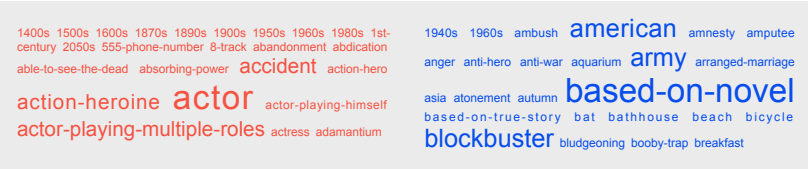


Fig. 3. Sample rating tag-clouds (left: rating 1, right: rating 5).

4.2 Average-based Rating

A very simple rating prediction strategy can be implemented by assuming that a given user u^* will rate a new movie m^* ($m^* \notin M_{u^*}$) according to the average rating that the movie received by all other users. We compute the average rating of movie m as

$$\bar{r}_m = \frac{1}{|U_m|} \sum_{u \in U_m} f_u(m) \,, \tag{2}$$

where $U_m = u \in U | m \in M_u$ is the set of users that have rated movie m, and $|U_m|$ is its cardinality. In this scheme, the predicted rating for movie m^* is the integer $r^* \in R$ that is nearest to $\bar{r}_{m^*}$.

4.3 Simple Tag-Cloud Comparison

In this scheme we guess the rating that user u^* would give to movie m^* by comparing the set of keywords K_{m^*} associated with the movie against the rating tag-clouds $T_{u^*,r}$ of user u^* for different ratings. We guess the rating r^* as the one corresponding to the tag-cloud (of user u^*) that most closely resembles the set of keywords K_{m^*}, as measured by the number of keywords that K_{m^*} shares with the tag-clouds of user u^* for different ratings:

$$\sigma(u^*, m^*, r) = |\{(k, n_k) \in T_{u^*,r} \,|\, k \in K_{m^*}\}| \,. \tag{3}$$

4.4 Weighted Tag-Cloud Comparison

In this hybrid scheme we try to take into account weights both at the keyword level (through their frequencies n_k) and at the tag-cloud level, though a measure of tag-cloud similarity. Given a new (in the sense of unrated) movie m^*, we consider the set of keywords K_{m^*} and introduce a notion of "similarity" between K_{m^*} and a given tag-cloud $T_{u,r}$. We define such a measure of similarity as:

$$\sigma(u, m, r) = \sum_{\{(k,n_k) \in T_{u,r} \,|\, k \in K_m\}} \frac{n_k}{\log(N_k)} \,, \tag{4}$$

that is we sum over all keywords which K_{m^*} and the tag-cloud $T_{u,r}$ have in common, and we weight each keyword k proportionally to its frequency n_k in the tag-cloud, and inversely proportional to the logarithm of its *global* frequency N_k, as commonly done in TFIDF term-weighting schemes.

We subsequently define the weighted average rating as

$$\bar{\sigma}(u, m) = \frac{1}{S(u, m)} \sum_{r \in R} r \, \sigma(u, m, r) \,, \tag{5}$$

where $S(u, m) = \sum_{r \in R} \sigma(u, m, r)$ is a normalization factor. Thus, $\bar{\sigma}(u, m)$ is an estimate of a user rating based on the weighted similarity between the set of movie keywords and the user's rating tagclouds (themselves weighted). This information can be used by itself, to guess a user rating, or it can be used to improve a prediction based on other techniques.

In our experiment we decided to use the rating $\bar{\sigma}(u, m)$, estimated from the tag-cloud similarity, to improve the simple rating estimate based on the per-movie average rating (see section 4.2). We combine the two estimates by computing their weighted average. That is, given a user u^* and a movie m^*, our estimate for the rating is

$$\sigma^*(u^*, m^*) = (1 - \gamma) \, \bar{r}_{m^*} + \gamma \, \bar{\sigma}(u^*, m^*) \,, \tag{6}$$

where $0 < \gamma < 1$ is a factor weighting the contribution of the two estimates. In our experiment we set $\gamma = 1/2$. We guess the rating r^* as the integer in R that lies closest to the weighted average $\sigma^*(u^*, m^*)$.

Of course, the above strategy can only be used when the set of keywords K_{m^*} associated with movie m^* is non-empty. If K_{m^*} is empty our implementation resorts to using the simple strategy of section 4.2 (equivalent to setting $\gamma = 0$ in Eq. 6).

5 Experiment and Results

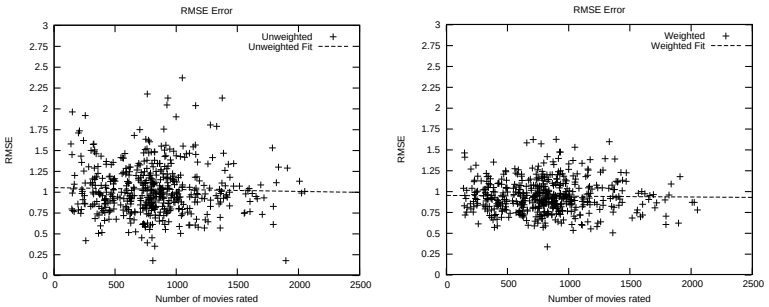
To test the algorithms presented earlier in Section 4, we extract a training set from the full Netflix data dump containing the ratings of 500 randomly chosen users. For each user, a test set made up from their last 100 ratings is removed from the training set so the accuracy of our algorithms can be tested. For each user, the root mean squared error (RMSE) is recorded, along with the percentage of exactly matched ratings. Given a set of predicted ratings $\{r_i\}$ and the corresponding set of actual ratings $\{r_i^*\}$, the RMSE is defined as:

$$\text{RMSE}(\{r_i\}, \{r_i^*\}) = \sqrt{\frac{1}{N}\sum_i (r_i - r_i^*)^2}\,. \tag{7}$$

A summary of the results follows:

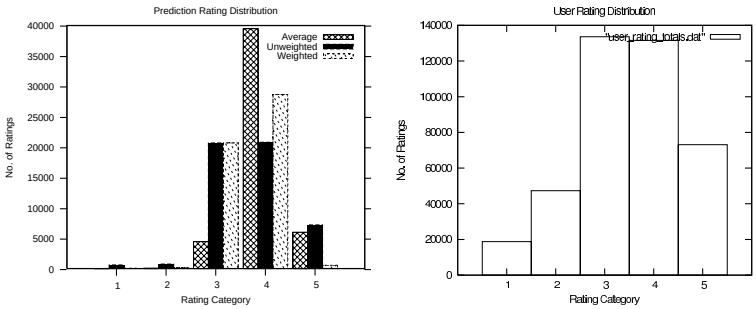
	Average Rating	Unweighted	Weighted
Correct	36.12%	44.15%	42.47%
Incorrect	63.99%	55.85%	57.53%
RMSE	1,131	1.074	0.961

The unweighted tag-cloud comparison does perform better than the naive average rating, with a moderate increase in the percentage of correctly rated movies. Using the weighted tag cloud comparison improves the RMSE, but with a slight drop in the fraction of exactly matched ratings. Figure 4 contains two scatter plots (unweighted and weighted tag-cloud comparison techniques) showing the RMSE for each user against the number of movies in their training set. These plots show two interesting features: (i) the weighted comparison technique has a smaller error range than the unweighted comparison (ii) the error rate seems to be independent of the number of movies rated. To visualise the distribution of predicted ratings for each of the algorithms, we present two histograms in Figure 5: one showing the distributions of the predicted ratings, and



(a) Unweighted RMSE breakdown (b) Weighted RMSE breakdown

Fig. 4. Scatter plots to show the level of accuracy for each rating technique in terms of the number of movies rated by the user.

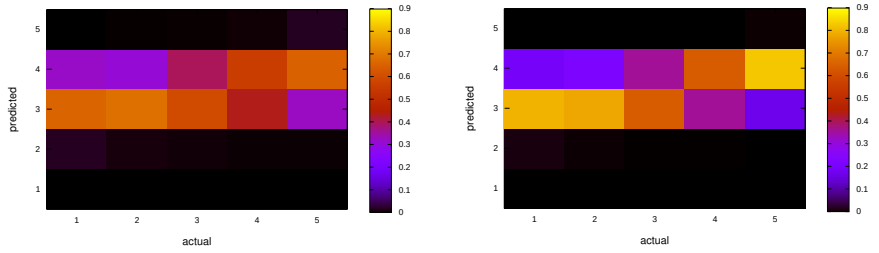


(a) Predicted rating distribution (b) Global rating distribution

Fig. 5. Histograms showing the number of predictions made in each rating category, and the overall rating distribution

one showing the global distribution of actual ratings. From these charts, it is clear that the rating categories 1 and 2 are being neglected.

In order to gain more insight into the behavior of our prediction schemes, we study the distribution of predicted ratings as a function of the actual rating. Fig. 6 shows the (color-coded) probability distribution of predicted ratings as a function of the actual movie rating, for the simple average-based scheme (left figure) and the weighted tag-cloud comparison scheme (right figure).



(a) Distribution of predicted ratings for average-based method

(b) Distribution of predicted rating for weighted tag-cloud comparison method

Fig. 6. Distribution of predicted ratings as a function of actual movie rating, for the simple average-based scheme (Figure 6(a)) and the weighted tag-cloud comparison scheme (Figure 6(b)). For each value of the actual rating (horizontal axis), a normalized histogram of the predicted ratings (vertical axis) was built, displaying how predicted values are distributed. Because of normalization, the sum of values along all columns is 1.

A perfect prediction scheme would appear as a unity matrix, with ones along the main diagonal and zeros elsewhere. Fig. 6 shows that both prediction schemes

behave poorly for low (1 and 2) and high (5) values of the actual rating, as both schemes predict intermediate ratings (3 and 4) with high probability, independent of the actual rating (bright rows in the plots).

We observe that the weighted tag-cloud scheme provides enhanced contrast throughout the rating range. For intermediate values of the actual rating (3 and 4) it improves significantly over the average-based scheme, with a better separation of the diagonal elements (3-3 and 4-4, correct predictions) over the off-diagonal ones, in particular over the elements corresponding to the incorrect predictions 3-4 and 4-3. For the highest actual rating (5) the weighted tag-cloud scheme features a distribution of predicted values which is more skewed towards high ratings, but on average it still fails to predict the correct rating. The same happens for low actual ratings (1 and 2), where the weighted tag-cloud scheme displays a distribution of predicted values which is more skewed towards low-values, but still fails to predict 1s and 2s with a significant probability.

In terms of future work, this evaluation shows that intermediate ratings are predicted rather well, and additional work is needed to make better prediction of extreme rating values, both high and low.

6 Conclusions and Future Work

In this paper, we have demonstrated that a movie recommendation system can be built purely on the keywords assigned to movie titles via collaborative tagging. By building different tag-clouds that express a user's degree of interest, a prediction for a previously unrated movie can be made based on the similarity of its keywords to those of the user's rating tag-clouds. With further work, we believe our recommendation algorithms can be improved by combining them with more traditional content-based recommender strategies. Since IMDB provides extensive information on the actors, directors, and writers of movies, as well as demographic breakdowns of the ratings, a more detailed profile can be constructed for each user. Also, our recommendation algorithms have not exploited any collaborative recommender techniques. Further research may show that rating tag-clouds are a useful and more efficient way to find neighbours with similar tastes.

References

1. Harith Alani, Yannis Kalfoglou, Kieron O'Hara, and Nigel Shadbolt. Towards a killer app for the semantic web. In *International Semantic Web Conference*, pages 829–843, 2005.
2. Tim Berners-Lee, Wendy Hall, James A. Hendler, Kieron O'Hara, Nigel Shadbolt, and Daniel J. Weitzner. A framework for web science. *Foundations and Trends in Web Science*, 1(1):1–134, 2006.
3. D. Billsus and M. Pazzani. User modeling for adaptive news access, 2000.
4. Chris Bizer, Richard Cyganiak, Jorg Garbers, and Oliver Maresch. D2RQ v0.5 - treating non-RDF relational databases as virtual RDF graphs. Technical report, Freie Universitat Berlin, 2006.

5. Elena Paslaru Bontas, Christoph Tempich, and York Sure. ONTOCOM: A cost estimation model for ontology engineering. In *Proceedings of the 5th International Semantic Web Conference (ISWC 2006)*, pages 625–639.
6. Robin Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370, 2002.
7. Del.icio.us: A social bookmarks manger homepage.
8. Scott Golder and Bernardo A. Huberman. The structure of collaborative tagging systems, Aug 2005.
9. Thomas Gruber. Ontology of folksonomy: A mash-up of apples and oranges, 2006.
10. Tony Hammond, Timo Hannay, Ben Lund, and Joanna Scott. Social bookmarking tools (i): A general review. *D-Lib Magazine*, 11, Apr 2005. 10.1045/april2005-hammond.
11. Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, January 2004.
12. J. A. Konstan, J. Reidl, A. Borchers, and J.L. Herlocker. Recommender systems: A grouplens perspective. In *Recommender Systems: Papers from the 1998 Workshop (AAAI Technical Report WS-98-08)*, pages 60–64. AAAI Press, 1998.
13. Cameron Marlow, Mor Naaman, Danah Boyd, and Marc Davis. Ht06, tagging paper, taxonomy, flickr, academic article, to read. In *HYPERTEXT '06: Proceedings of the seventeenth conference on Hypertext and hypermedia*, pages 31–40, New York, NY, USA, 2006. ACM Press.
14. Adam Mathes. Folksonomies – Cooperative Classification and Communication Through Shared Metadata, December 2004. http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html.
15. Stuart E. Middleton, Nigel R. Shadbolt, and David C. De Roure. Ontological user profiling in recommender systems. *ACM Trans. Inf. Syst.*, 22(1):54–88, 2004.
16. Peter Mika. Ontologies are us: A unified model of social networks and semantics. In *International Semantic Web Conference*, LNCS, pages 522–536. Springer, 2005.
17. Netflix homepage.
18. Netflix prize homepage.
19. Satoshi Niwa, Takuo Doi, and Shinichi Honiden. Web page recommender system based on folksonomy mining for itng '06 submissions. In *ITNG '06: Proceedings of the Third International Conference on Information Technology: New Generations (ITNG'06)*, pages 388–393, Washington, DC, USA, 2006. IEEE Computer Society.
20. Ben J. Schafer, Joseph A. Konstan, and John Riedi. Recommender systems in e-commerce. In *ACM Conference on Electronic Commerce*, pages 158–166, 1999.
21. I. Schwab, A. Kobsa, and I. Koychev. Learning user interests through positive examples using content analysis and collaborative filtering, 2001.
22. S. Staab, S. Santini, F. Nack, L. Steels, and A. Maedche. Emergent semantics. *IEEE Intelligent Systems*, 17(1):78–86, 2002.
23. Vander Wal. T. Folksonomy definition and wikipedia, November 2005.
24. The internet movie database - wikipedia entry.
25. The internet movie database (IMDB) homepage.

Revision and Co-revision in Wikipedia*

Detecting Clusters of Interest

Ulrik Brandes and Jürgen Lerner**

Department of Computer & Information Science, University of Konstanz

Abstract. The online encyclopedia Wikipedia gives rise to a multitude of network structures such as the citation network of its pages or the co-authorship network of users. In this paper we analyze another network that arises from the fact that Wikipedia articles undergo perpetual editing. It can be observed that the edit volume of Wikipedia pages varies strongly over time, often triggered by news events related to their content. Furthermore, some pages show remarkably parallel behavior in their edit variance in which case we add a co-revision link connecting them. The goal of this paper is to assess the meaningfulness of the co-revision network. Specific tasks are to understand the influence of normalization (e.g., correlation vs. covariance) and to determine differences between the co-revision network and other relations on Wikipedia pages, such as similarity by author-overlap.

1 Introduction

Wikipedia[1] is a Web-based collaborative authoring environment, where anyone on the Internet can create and modify pages about encyclopedic topics. Since its creation in 2001, Wikipedia enjoys increasing popularity. At the end of 2006, Wikipedia has more than five million articles—about 1.5 million alone in the English Wikipedia—and grows by several thousand articles per day.[2]

There are several fundamental differences between Wikipedia pages and traditional articles (e. g., articles written for scientific journals or conference proceedings or entries in printed encyclopedias). Firstly, Wikipedia articles are written without centralized supervision, i. e., there are no editors deciding over which topics are treated and how much space is reserved for a certain entry. Furthermore, articles can be included and edited without a prior review process. Secondly, Wikipedia pages are written by up to thousands of authors, potentially having different education, knowledge, interests, and opinions on the topic. Thirdly, Wikipedia pages are never finished but undergo perpetual and frequent editing.

In this paper we focus on the latter two properties. Thereby, we have two goals in mind: The first is to better understand the content-generation process

* Research supported by DFG under grant Br 2158/2-3

** Corresponding author, lerner@inf.uni-konstanz.de

[1] http://www.wikipedia.org/

[2] http://stats.wikimedia.org/

of Wikipedia by tackling questions such as what is the typical edit volume of a Wikipedia page and how does it evolve over time, which pages are frequently revised during the same time periods, and which pages have many common authors. The second goal is to exploit these two properties to define similarity between pages: a *co-author similarity* measuring how much the author sets of two pages overlap and a *co-revision similarity* measuring to what extend two pages are edited during the same time intervals. Here we want to tackle how these measures have do be defined (e. g., which normalization is appropriate) such that meaningful and non-trivial similarity is obtained.

1.1 Related Work

Wikipedia has been established in 2001 to collectively create an encyclopedia. Maybe due to its size, popularity, and relevance for understanding new forms of collective knowledge creation, Wikipedia receives increasing interest in research. A study carried out by *Nature* in 2005 suggests that the accuracy of Wikipedia articles about scientific topics comes close to the accuracy of their counterparts in the *Encyclopaedia Britannica* [3]. Viégas *et al.* [7, 8] proposed a *history flow* approach for the visual analysis of the page history. A difference to our paper is that [7, 8] focus on the text of the page and we on the revision behavior. Work in [6] analyzes the information quality of Wikipedia articles by defining and measuring attributes such as authority, completeness, and volatility. The growth of Wikipedia is described in, e. g., [9, 1], whereas [4] analyze category-membership of articles. Other papers (e. g., [2, 5]) use the collection of Wikipedia articles to improve machine learning techniques for text categorization and detection of semantic relatedness of terms.

1.2 Input Data

Wikipedia makes its complete database (containing all versions of every article since its initial creation) available in XML-format.[3] The files containing the complete history of all pages can be extremely large. For instance, the complete dump for the English Wikipedia unpacks to more than 600 gigabytes (GB).[4] Wikipedia makes also available so-called stub-files. These files contain meta-data about every revision but not the text and are still quite large. For the present study we used the stub-file for the English Wikipedia (which is the largest one) from the 2006-11-30 dump with a size of 23 GB. (Note that this dump includes some revisions from December 2006, since it takes several days to create it.) More precisely, we used only the information "who performed when a revision to which page." Parsing the XML-document has been done with a Java implementation of the event-based SAX interfaces[5] which proved to be very efficient for parsing such huge files. Constructing the whole document tree, as

[3] `http://download.wikimedia.org/`
[4] `http://meta.wikimedia.org/wiki/Data_dumps`
[5] `http://www.saxproject.org/`

this is normally done by DOM parsers,[6] would simply be impossible (at least very inefficient and/or requiring uncommonly huge memory), given the file sizes. In the whole paper we consider only pages from the main namespace (i. e., we do not consider, discussion pages, user pages, user-talk pages, etc.). Some computations (especially in Sects. 3 and 4) are performed only for those pages that have more than 2000 edits. There are 1,241 pages in the 2006-11-30 dump satisfying this criteria (compare the remarks at the beginning of Sect. 3).

2 Statistics on Single Pages

The time-stamp of a revision denotes the exact second when this edit has been inserted in Wikipedia. When comparing the edit volume of Wikipedia pages over time, however, we adopt a much coarser point of view and consider their *weekly* number of edits. The decision "one week" is in a certain sense arbitrary and exchangeable by longer or shorter intervals of time. Furthermore, this decision certainly has an influence on the co-revision network defined in Sect. 3. However, we have chosen a week as this marks how people normally organize their work. Thus, a page that undergoes every week the same number of revisions but that is edited more often on week-ends than during the week is not considered to have a varying edit volume.

A second difficulty arises from the fact that Wikipedia pages are not all created at the same time. For instance, the page `2006 Israel-Lebanon conflict` does not even have the possibility to exist before 2006 (assuming that no author tries to predict the future). While this does not matter when we consider single pages, the problem has to be solved how to compare the edit volume of two pages that have different lifetimes. A first convention is to ignore the time when only one page existed, a second is to consider the longer time interval and take the point of view that pages received zero edits during the time when they did not yet exist. We will adhere to the second convention (more precisely we always consider the time from January 2001 until December 2006) for two reasons. Firstly, we do not want to ignore the fact that some pages are created earlier than others, as this already marks a difference between them. Secondly, measures like the covariance of the edit volume of two pages (used in Sect. 3) are hard to compare if we take them over different numbers of intervals (considering only the lifetime of the youngest Wikipedia page is obviously not an option, as this is simply too short).

Let p be a Wikipedia page and let $r_i(p)$ denote the number of revisions on page p in week i, where the weeks are assumed to be indexed with $i = 1, \ldots, K$. The value $R(p) = \sum_{i=1}^{K} r_i(p)$ is the total number of revisions on page p, $r_{\max}(p) = \max_{i=1,\ldots,K} r_i(p)$ the maximum number of weekly edits on page p, and $\mu_r(p) = R(p)/K$ the *mean* value (average number of edits per week). Furthermore, $\sigma_r^2(p) = \sum_{i=1}^{K}(r_i(p) - \mu_r(p))^2/K$ is the *variance* of p's edit volume (denoting the expected squared difference to its mean value) and $\sigma_r(p) = \sqrt{\sigma_r^2(p)}$ the *standard deviation*.

[6] `http://www.w3.org/DOM/`

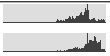
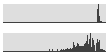
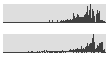
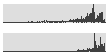
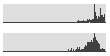
For a page p, let $A(p)$ denote the set of authors (logged-in or anonymous) that performed at least one edit to p and let $a(p) = |A(p)|$ denote the size of p's author set. Authors that are logged-in are identified by their username. The anonymous authors are identified by the IP-address of the computer from which they made the contribution. A problem arising from the inclusion of anonymous authors is that the same person might be logged-in using different IP-addresses, in which case we would count him/her several times. We have chosen to include anonymous authors since we observed that some of them make valuable and frequent contributions. Nevertheless, interpretation of the numbers of authors should take it into account that they probably contain many duplicates.

It is straightforward to aggregate values over a set of pages. For instance, if P is the set of all Wikipedia pages (from the main namespace), then $r_i = \sum_{p \in P} r_i(p)$ is the edit volume of Wikipedia in week i (for $i = 1, \ldots, K$).

2.1 Most-edited Pages

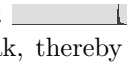
Table 1 lists the ten pages with the maximal average number of edits per week. Since we took the average over the same number of weeks for all pages (see above), these are also the Wikipedia pages having the highest number of edits in total. The last row in Table 1 denotes the values obtained by summing up the weekly edit number over all pages.

Table 1. The ten pages with the maximal average number of edits per week (real numbers are rounded to integer). The diagrams in the second column show the number of edits per week. The horizontal time-axis in these diagrams is the same for all pages (i. e., it goes over six years). In contrast, the vertical axis is scaled to unit height, so that the same height means a different number for different pages (maximum number of edits per week, corresponding to unit height, is denoted in the third column).

title(p)	$r_i(p)_{i=1,\ldots,K}$	$r_{\max}(p)$	$\mu_r(p)$	$\sigma_r(p)$	$a(p)$
George W. Bush		992	105	164	10,164
Wikipedia		630	70	115	9,275
United States		635	54	91	5,926
Jesus		735	50	87	4,302
2006 Israel-Lebanon conflict		3,679	48	**319**	2,755
Adolf Hitler		415	46	70	5,218
World War II		507	45	77	5,260
Wii		998	44	114	4,585
RuneScape		505	43	85	4,650
Hurricane Katrina		3,946	41	**246**	4,527
all pages		942,206	198,179	281,802	5.2 million

The topics of the most-edited pages span a broad range from people over countries and historic events to online games and a game console. However, the focus of this paper is on the differences and similarities in the revision characteristics of pages rather than their topics.

The numbers counting edits and authors appear to correlate quite well. A slight deviation from this rule is the page `2006 Israel-Lebanon conflict` having a smaller number of authors than other pages with so many edits (this page has only rank 68 in the list of pages having the most authors). The correlation (see the definition of correlation in Sect. 3.2) between the number of authors and the number of revisions (computed over all pages having at least 2,000 revisions, compare Sect. 3) is 0.88. Thus, pages with many authors indeed tend to have many revisions and vice versa.

Much more significant are the differences in the standard deviation (and thus also in the variance) of the edit volumes. For instance, the high variance of the pages `2006 Israel-Lebanon conflict` and `Hurricane Katrina` (printed in bold in Table 1) is probably due to the fact that interest in these pages is triggered by the events they describe. While theses two pages did not exist before the respective event, it turns out later that some pages that existed much earlier received also a high increase in interest at the respective time points. The edit plots of `2006 Israel-Lebanon conflict`  and `Hurricane Katrina`  show both a very narrow peak, thereby making the high variance visible.

The edit plots also reveal characteristics of other pages that are not so extremely reflected in the variance. For instance, the edit volume of `George W. Bush`  first increases. Then it suddenly drops and remains rather constant at a certain level. Probably the reason is that this page is a frequent target of vandalism and was the first page that became protected (compare [8]) resulting in a decrease in the number of edits.

The aggregated number of weekly edits for all Wikipedia pages is generally increasing, so that Wikipedia as a whole is more and more edited. Not surprisingly, the aggregated plot  is much smoother than those of single pages.

3 The Co-revision Network

Some Wikipedia pages appear to have quite parallel edit volumes, so that interest in these pages is raised at the same time. In this section we analyze the network arising if we consider two such pages as similar. In doing so we have two goals in mind: firstly to understand better which kind of pages are frequently *co-revised* and secondly to assess whether co-revision helps us to establish meaningful and non-trivial similarity of pages. Special emphasis is given in developing appropriate normalization for the edit plots and similarity values. In Sect. 4 we compare the co-revision network with the network derived from author-overlap.

In this section we are confronted with the problem that computing the co-revision network on all Wikipedia pages leeds to unacceptably high running times and memory consumption, since the matrix encoding the co-revision for all pairs of pages is obviously quadratic in their number (1.5 million in the main namespace). One possibility to make the computation fast enough (and still to analyze hopefully all interesting pages) is to reduce the number of pages

by considering only those that received a minimum number of edits. (A second advantage when doing so is that several normalizations become more stable since we do not divide by numbers that are too small.) In Sects. 3 and 4 we considered only those pages from the main name space that have at least 2000 edits (1,241 pages satisfy this criterion).

3.1 Covariance

The (*weekly*) *edit covariance* or (*weekly*) *revision covariance* of two pages p and q is defined to be

$$cov_r(p,q) = \sum_{i=1}^{K} [r_i(p) - \mu_r(p)] \cdot [r_i(q) - \mu_r(q)]/K \ .$$

The covariance is symmetric, i. e., $cov_r(p,q) = cov_r(q,p)$.

To get an overview of the covariance network we applied a quite simple method, described in the following, which will also be applied to the (much more useful) correlation network (see Sect. 3.2) and author-overlap network (see Sect. 4). Let $Cov = (cov(p,q))_{p,q \in P}$ denote the matrix containing the covariance values for all pairs of pages and let k be an integer. The graph G^k_{cov} is the graph whose edges correspond to the k entries in Cov having the highest values and whose vertices are the pages incident to at least one edge. The graph resulting from the 50 strongest edges is shown in Fig. 1. This network contains only the pages with the highest variances. In conclusion, covariance does not seem to yield insightful similarity of pages.

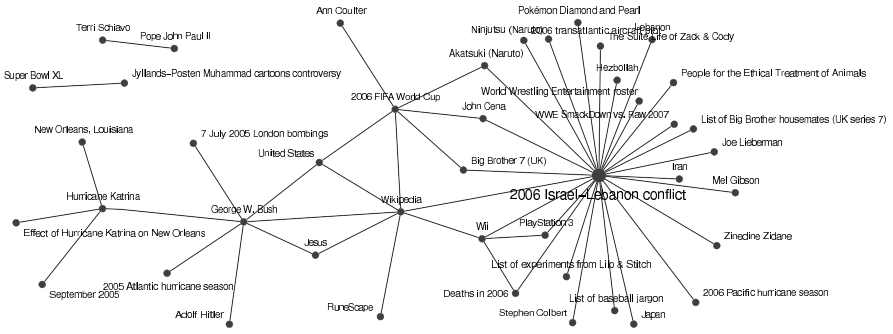


Fig. 1. Image of the graph G^{50}_{cov}. The central vertex corresponding to `2006 Israel-Lebanon conflict` is shown larger. This page has the highest variance of all Wikipedia pages and dominates the covariance network.

3.2 Correlation

Since edit covariance is highly influenced by the variance of the pages' edit volume it is reasonable to normalize these values. The (*weekly*) *edit correlation* or (*weekly*) *revision correlation* of two pages p and q is defined to be

$$corr_r(p,q) = \frac{cov_r(p,q)}{\sigma_r(p)\sigma_r(q)} \ .$$

The correlation is symmetric (i. e., $corr_r(p,q) = corr_r(q,p)$) and in $[-1,1]$.

The two pages with the highest correlation are `Hurricane Katrina` and `Effect of Hurricane Katrina on New Orleans` which reach a correlation of 0.97 (i. e., close to the maximum). As for the covariance network we construct the graph G^{50}_{corr} (see Fig. 2) from the 50 edges corresponding to the largest correlation values.

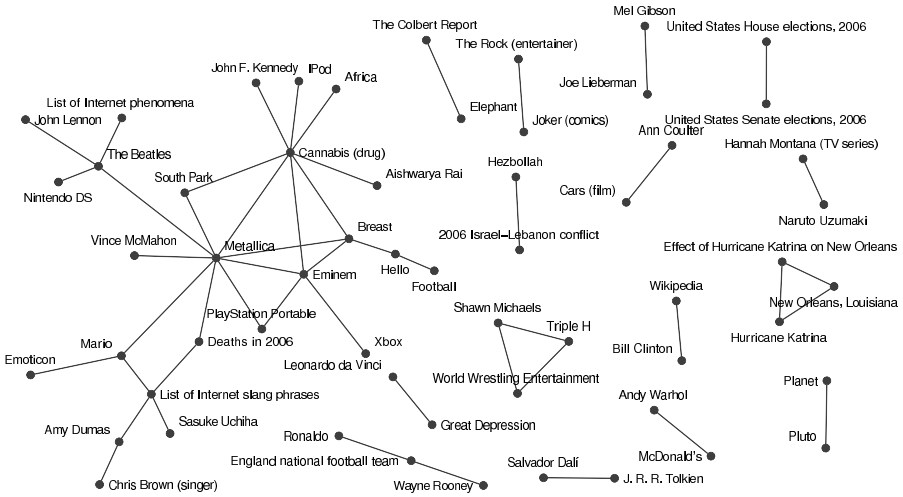
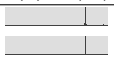
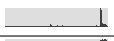
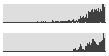
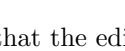


Fig. 2. Graph constructed from the 50 edges with highest correlation values.

Some of these correlations appear to be meaningful, others not. For instance it is reasonable that the three pages related to "Hurricane Katrina", the two pages related to the "2006 Israel-Lebanon conflict", and also the two pages `Pluto` and `Planet` are frequently revised at the same time (Pluto lost its status as a planet in 2006). Indeed, as Table 2 shows, some of the associated edit plots look remarkably similar, although they reach very different maximal values.

On the other hand, some correlations seem to be quite arbitrary. To understand why these pages are nevertheless so highly correlated we look at prominent members of the largest connected component in the left part of Fig. 2 and show

Table 2. Edit plots of selected pages showing a high correlation (compare Fig. 2).

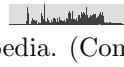
title(p)	$r_i(p)_{i=1,\ldots,K}$	$r_{\max}(p)$
`Hurricane Katrina`		3,946
`Effect of Hurricane Katrina on New Orleans`		1,099
`New Orleans, Louisiana`		533
`2006 Israel-Lebanon conflict`		3,679
`Hezbollah`		681
all pages		942,206
`Metallica`		138
`Cannabis (drug)`		221
`South Park`		212
`Eminem`		313

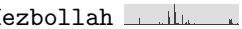
their edit plots in Table 2. What can be observed is that the edit plots of these pages do not look very special with respect to the aggregated edits of all pages. Especially the plots of `Cannabis (drug)` and `Metallica`, which are the most connected pages in Fig. 2, are very similar to the aggregated plot. So our current hypothesis is that some pages are just similar with respect to edit correlation because they are edited like the average Wikipedia page.

In conclusion, the similarity values derived by correlation of the weekly number of edits are much better than those derived from covariance. However, while a high correlation might point to a meaningful connection between the pages it is not necessarily so. The major drawback of correlation seems to be that pages that are edited as the average Wikipedia are assigned high similarity values, independent on whether they treat related topics. In the next subsection we attempt to filter this out.

3.3 Relative Edit Volume

Considering the strongly skewed aggregated edit volume of Wikipedia and having in mind the remarks at the end of the previous subsection, it may be worthwhile to consider the *relative edit volume* of individual pages, i. e., the percentage that a specific page receives from the weekly edits done in the entire Wikipedia. So, let $r_i(p)$ denote the number of edits of page p in week i and r_i denote the total number of edits on all Wikipedia pages in week i. Then $r_i(p)/r_i$ is called the *relative number of edits* of page p in week i. This yields the measures *relative edit covariance* and *relative edit correlation*, compare Sects. 3.1 and 3.2.

The plots showing the relative edit volume reveal some interesting characteristics of the pages. For instance the page `George W. Bush`  receives high (relative) interest already in the early days of Wikipedia. (Compare the plot showing the absolute number of edits  which begins to rise later.) Even more extreme is the difference between the relative edit plot of `Rammstein`  showing a single peak at the beginning of Wikipedia and its absolute edit plot  which indicates more interest in later years.

The comparison between the relative and the absolute edits also provides a distinction between pages that are solely edited during a certain event and pages that only show a strong increase in interest during events. For instance, the absolute edit plots of `2006 Israel-Lebanon conflict` and `Hezbollah` are very similar. On the other hand, their relative plots reveal that the page `2006 Israel-Lebanon conflict`  is relative to the whole Wikipedia still focused on that event, whereas the page `Hezbollah`  receives the most edits (relative to the whole Wikipedia) much earlier.

Motivated by such examples we thought that *relative covariance* and *relative correlation* would yield similarity values which are more reliable than their counterparts derived from the absolute edit volume. However, it turned out that this is not the case. Instead, both the relative covariance and the relative correlation are dominated by a few edits in the early days of Wikipedia when the aggregated number of edits was by orders of magnitude smaller than in later years.

In conclusion, normalizing the edit volumes by the aggregated number of edits seems to be a natural way to prevent that pages become similar just because they behave like the average page (compare Sect. 3.2). However, since the aggregated edit volume  is highly skewed this involves division by very small numbers (compared to the largest ones) and thus yields a highly unstable method. It is an issue for future work to develop a more appropriate normalization.

4 The Co-author Network on Pages

Some Wikipedia pages have thousands of authors. In this section we consider similarity of pages derived from overlapping author sets. As in Sect. 3 we have two goals in mind: firstly to understand better which kind of pages are frequently *co-authored* and secondly to assess whether co-authoring helps us to establish meaningful and non-trivial similarity of pages. In addition we want to compare the co-revision and co-author network. The term "co-author network" often denotes networks of authors (in contrast, we construct a network of pages) connected by commonly written articles. However, in this section we consider only the network of Wikipedia pages resulting from overlapping author sets.

A first possibility is to define similarity of pages by simply counting the number of common authors, i. e., taking the values $a(p, q) = A(p) \cap A(q)$ as a measure of author overlap between two pages p and q. (We remind that $A(p)$ denotes the set of authors (logged-in or anonymous) of a page p and $a(p) = |A(p)|$ denotes the number of its authors.)

The two pages with the highest number of common authors (namely $1,074$) are `George W. Bush` and `Wikipedia` which are also the two pages having the largest number of authors (both roughly $10,000$). As for the co-revision network (compare Figs. 1 and 2) we construct the graph arising from the 50 strongest values in $a(p, q)$, see Fig. 3. As it could be expected, the un-normalized co-authoring similarity $a(p, q)$ is highly biased towards pages with large author sets

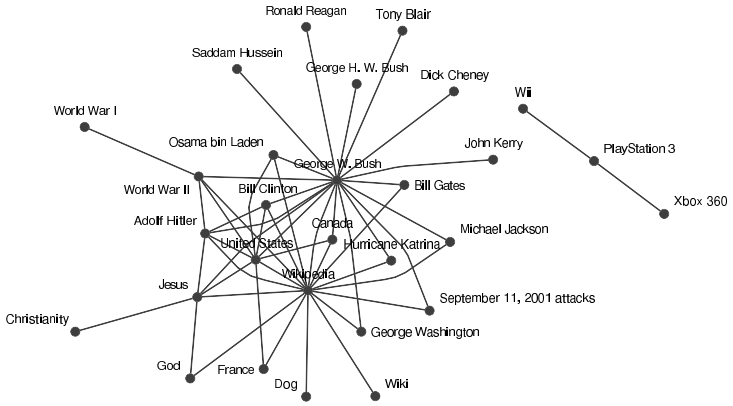


Fig. 3. Graph whose edges are the 50 pairs having the highest numbers of common authors. This graph is dominated by pages having the largest numbers of authors.

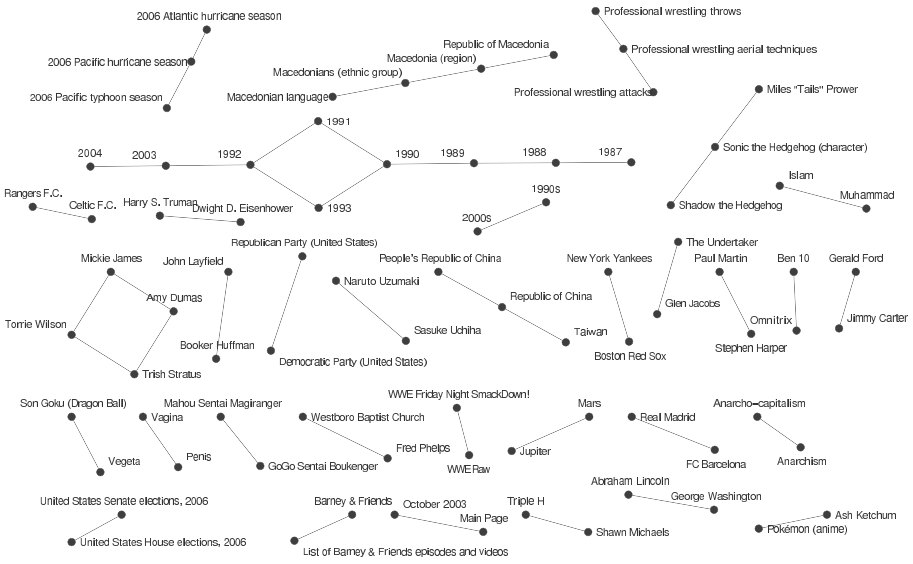


Fig. 4. Graph whose edges are the 50 pairs having the highest normalized numbers of common authors. The connected components correspond in most cases well to pages with similar topics.

although a reasonable cluster containing three pages about game consoles is identified.

Similar to the normalization of covariance to correlation we normalize the number of common authors by dividing with the geometric mean of the numbers of authors:

$$a_{\cos}(p,q) = \frac{a(p,q)}{\sqrt{a(p)a(q)}} \ .$$

(The notation $a_{\cos}$ has been chosen since this measure is the cosine of the angle between the characteristic vectors of the two author sets.) The normalized number of common authors $a_{\cos}(p,q)$ ranges between zero and one. The highest value is between the two pages `2006 Atlantic hurricane season` and `2006 Pacific hurricane season` (reaching a value of 0.27). The 50 strongest values give rise to the graph shown in Fig. 4. The connected components of this graph appear to be quite reasonable as they normally consist of pages treating strongly related topics.

It is remarkable that the graphs stemming from correlation in the number of weekly edits (Fig. 2) and from normalized author overlap (Fig. 4) are almost disjoint (an exception are the United States elections 2006). Indeed, from a qualitative point of view the former seems to connect pages that are related to the same events and the latter to connect pages having related topics. Co-revision and co-authoring also are quite independent from a quantitative point of view: the correlation between the values $a_{\cos}(p,q)$ and $corr_r(p,q)$ (see Sect. 3.2) is 0.18 and the correlation between the values $a(p,q)$ and $cov_r(p,q)$ (see Sect. 3.1) is 0.39. (Note that we performed this computation only for all pairs of pages that have more than 2000 edits, so that the results might not generalize to the set of all Wikipedia pages.) These low correlation values indicate a rather weak dependence between co-revision and co-authoring similarity. The somewhat higher correlation between the un-normalized versions is probably due to the fact that pages with higher covariance normally have a higher total number of edits and, thus, more authors (compare Table 1). In conclusion, co-revision and co-authoring seems to be quite different—at least for the pages with many edits.

5 Discussion and Future Work

In contrast to traditional articles, Wikipedia pages have huge author sets and are permanently edited. This paper analyzes these two properties and achieves some initial findings.

The plots of the edit volume of pages over time reveal some interesting characteristics: For instance, some pages show an overall increase in interest; others are mostly edited during certain events. Furthermore, the plots of two pages shown simultaneously reveal whether these pages are edited in parallel.

When analyzing co-revision similarity, it became evident that correlation performs much better than covariance, since the latter is biased towards pages of very high variance. The similarity derived from correlation seems to be quite

meaningful for some pages and rather arbitrary for others. Our current hypothesis is that pages that are edited as the average Wikipedia page receive quite large correlation values, independent on whether they are somehow related or not. Attempts to filter this out by considering the relative edit volumes failed due to the highly skewed distribution of the aggregated edit volume. It is an open problem to develop a better normalization strategy.

The normalized version of the co-authoring similarity seems to yield quite meaningful associations. Co-authoring similarity and co-revision similarity appear to be rather unrelated, so that co-revision might point to complementary relatedness of pages. Furthermore, co-revision can be applied to relate pages written in different languages whose author sets are normally non-overlapping.

Further issues for future work include developing appropriate clustering algorithms for the co-revision or co-authoring networks, analyzing how the content of a page changes after it received a peak of interest (pages such as `Hezbollah` or `New Orleans, Louisiana`, compare Table 2), and comparing co-revision and co-authoring to other network structures such as hyperlinks pointing from one page to another or common membership in categories.

References

1. L. S. Buriol, C. Castillo, D. Donato, S. Leonardi, and S. Millozzi. Temporal analysis of the wikigraph. In *IEEE/WIC/ACM International Conference on Web Intelligence (WI'06)*, pages 45–51, 2006.
2. E. Gabrilovich and S. Markovitch. Overcoming the brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge. In *Proceedings of the 21'st National Conference on Artificial Intelligence*, 2006.
3. J. Giles. Internet encyclopaedias go head to head. *Nature*, 438:900–901, 2005.
4. T. Holloway, M. Božičević, and K. Börner. Analyzing and visualizing the semantic coverage of Wikipedia and its authors. arXiv:cs/0512085.
5. M. Strube and S. P. Ponzetto. WikiRelate! computing semantic relatedness using wikipedia. In *Proceedings of AAAI'06*, 2006.
6. B. Stvilia, M. B. Twindale, L. C. Smith, and L. Gasser. Assessing information quality of a community-based encyclopedia. In *Proceedings of the International Conference on Information Quality*, 2005.
7. F. B. Viégas, M. Wattenberg, and K. Dave. Studying cooperation and conflict between authors with history flow visualizations. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 575–582, 2004.
8. F. B. Viégas, M. Wattenberg, J. Kriss, and F. van Ham. Talk before you type: Coordination in Wikipedia. In *Proceedings of HICSS 40*, 2007.
9. J. Voss. Measuring Wikipedia. In *Proceedings of the International Conference of the International Society for Scientometrics and Informetrics*, 2005.

Organizing Resources on Tagging Systems using T-ORG

Rabeeh Abbasi[1], Steffen Staab[1] and Philipp Cimiano[2]

[1]ISWEB, University of Koblenz-Landau, Germany
http://isweb.uni-koblenz.de
{abbasi,staab}@uni-koblenz.de
[2]Institute AIFB, University of Karlsruhe, Germany
http://www.aifb.uni-karlsruhe.de/WBS
pci@aifb.uni-karlsruhe.de

Abstract. Tagging systems (or folksonomies) like Flickr or Delicious are expanding tremendously. More and more resources are being added to them. As the resources present on these system increase in amount, it becomes difficult to explore these resources. For this purpose, we present a system T-ORG, which provides a mechanism to organize these resources by classifying the tags (or keywords) attached to them into predefined categories. Supervised classification in this case seems infeasible; therefore we also propose a new classification algorithm T-KNOW that does not require training data. For our experiments, we have downloaded images and their tags from groups present on Flickr website and then classified these tags into different categories. We have used Cohen's Kappa and F-measure to evaluate the classification results of T-KNOW. Results are encouraging and show that T-ORG can be used to explore resources in an effective manner.

Keywords: Tags Classification, Tagging Systems, Folksonomies, Semantic Web, Cohen's Kappa

1 Introduction

More people are being attracted to tagging systems like Flickr[1] or Delicious[2] because of the number of benefits they provide. For example, they are easy to use and do not require any specific skills. Users can search and browse resources using the tags (keywords) attached to the resources. They also provide "Tag Clouds" to browse resources. In a "Tag Cloud", frequently used tags are displayed in large text. Despite of all these benefits, sometimes it might become difficult for a user to browse particular types of resources. Just consider the scenario in which a user wants to explore vehicle images. Considering current searching and browsing facilities

[1] http://www.flickr.com/

[2] http://del.icio.us/

provided by these systems, it seems difficult to browse only a particular kind of resources. This problem of exploring resources of a particular type can be solved by additional classification of resources. Classifying resources into predefined categories can provide a mechanism to explore a particular type of resources present on a tagging system. It can improve the user experience and can add more benefit to existing tagging systems. Manually classifying the resources available on a tagging system is not feasible, because of the tremendous amount of data present and being added to it. Therefore some kind of system is required which can classify resources on a tagging system into some categories without any supervision.

We have explored means to automatically organize tags into hierarchies in order to explore resources in tagging systems and to provide better browsing experience to the user. For this purpose, we have developed a system T-ORG (Tag-ORGanizer), which classifies the resources of a tagging system into predefined categories and helps in browsing a particular type of resources available on tagging system. The classification of resources is based on the classification of tags attached to these resources. If a resource has two tags having two different categories, then the resource is classified as both of these categories. For example, if a resource has tags “Paris” and “Peugeot” and these tags are classified as “Location” and “Vehicle” respectively, then the resource is placed in both of these categories (i.e. Location and Vehicle). Tag classification can help a user to use tags on a tagging system in a more organized way. For example, instead of representing different tags in a tag cloud, sometimes it could be more useful, if a “Tag Cloud” displays the abstract tags (i.e. categories) and when a user clicks an “abstract tag”, its subsequent tags are displayed. In such way, a user can explore different type of tags (and hence resources) available on a tagging system, which might not be possible with a simple “Tag Cloud”.

The core of T-ORG is its classification method T-KNOW (Tag classification using KNowledge On the Web). It is based on an unsupervised mechanism for classifying tags in folksonomies. T-KNOW uses Google for finding categories of tags; therefore it does not require any training and can be used for unsupervised classification of tags (like [3]). It classifies the tags into categories using its pattern library, categories extracted from a given ontology and Google search results. As there might be several results returned by Google against a query posed by T-KNOW, a method is required to select best results on the basis of the similarity between tagging and search results. T-KNOW uses the context of the tag to measure the similarity between Google search results and the tag. We also propose four methods of selecting the context of a tag.

2 Process of T-ORG (Tag-ORGanizer)

The purpose of T-ORG is to organize resources by classifying their tags into categories. This process is done by selecting concepts from single or multiple ontologies related to the required categories and then pruning and refining these ontologies. These concepts are considered as categories into which the tags are classified. Figure 1 shows the overall process of T-ORG while each step is described below.

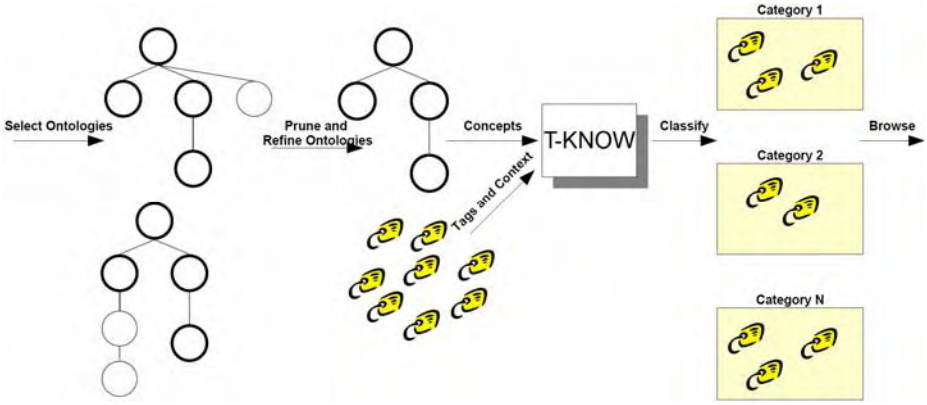


Fig. 1. Process of T-ORG

Selecting Ontology

The user of T-ORG has to decide about the categories into which the resources are to be classified. The user selects ontologies relevant to the required categories. Concepts from these ontologies are used as categories. For example to browse through the images of vehicles at Flickr, one would select a vehicle ontology. Currently this step is done manually in T-ORG.

Pruning and Refining Ontology

After selecting ontologies, they must be pruned and refined for the desired categories. Only those concepts from these ontologies are considered which have some relation to the required categories. Unwanted concepts are pruned. Redundant and conflicting concepts are refined. Missing concepts are also added into the given ontology. For example to include the images of a “draisine”, one might have to add this concept into a given vehicle ontology. Once the ontology is pruned and refined, its concepts are used as categories. Currently this step is also done manually in T-ORG.

Applying T-KNOW for Classifying Tags

Classifying the tags is a major step in the process of T-ORG. Once the ontology is selected, pruned, and refined, and categories are extracted from this ontology, then these categories and the context of tags are used for classification. Once all tags are classified into categories, each category is subsumed by its parent category, for example, every tag classified as Train, Bulldozer or Bus is finally classified as Vehicle. Section 3 describes the detailed process of classifying tags using T-KNOW.

Browsing the Resources

After classifying each tag, resources may be browsed according to the categories assigned to their tags. The browser may use information of resources to display them in categories, so that the user can browse particular type of resources present in these categories.

3 Tag Classification using knowledge On the Web

T-KNOW uses lexico-syntactic patterns and Google APIs for finding the appropriate categories of the tags. Given a list of tags and categories, T-KNOW classifies these tags into categories. It builds queries by combining linguistic patterns (Hearst Patterns [7] and a few more [3]) and the categories and then searches these queries on Google using Google API. The process of classifying tags using T-KNOW is shown in Figure 2. In what follows, we describe in more detail the steps shown in Fig. 2.

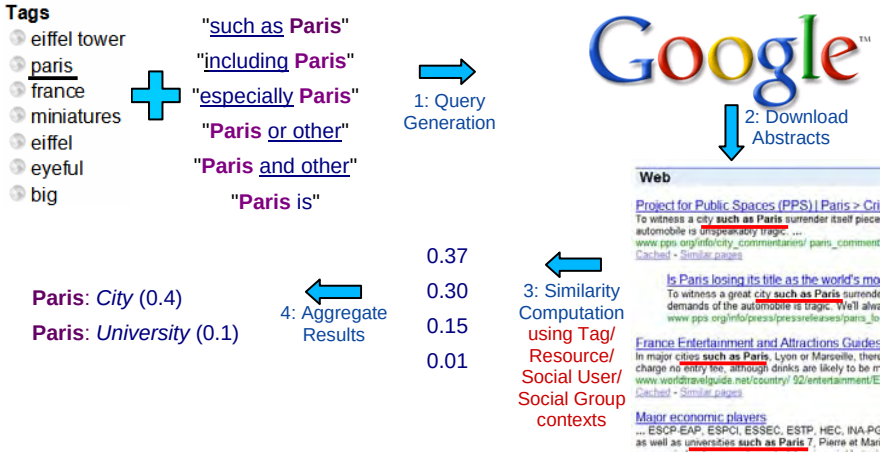


Fig. 2. Process of T-KNOW

Assume a tag like "Paris" is to be classified in a context as depicted in Figure 4

Step 1: Queries are generated by concatenating the tag and the clues, e.g. "such as **Paris**" is a query generated by combining the clue "such as" and the tag "Paris"

Step 2: The queries are searched using the Google API and abstracts of search results are downloaded, e.g. "To witness a **city** such as *Paris* surrendered itself…" is a search result abstract downloaded for the query "such as Paris"

Step 3: The similarity between each abstract and context (described in Section 3.1) of tag is computed, e.g. between the abstract "To witness a city such as Paris…" and context of the tag "Paris" (eiffel tower, france, miniatures…). If similarity is above a certain threshold value, then depending upon the clue used, the abstract is matched against the pattern, e.g. the abstract "To witness **city** such as *Paris*…" is matched against the Hearst pattern [7] "**CONCEPT** such as (*INSTANCE*,?)+ ((and|or) INSTANCE)", where CONCEPT is the expected category and INSTANCE is the tag. Hence "City" is extracted as an expected category of the tag "Paris" from this abstract.

Step 4: The results are aggregated and the category having highest similarity with the tag's context is returned, e.g. for the tag "Paris" the category "City" is

returned, because it has higher similarity than the other category e.g. "University"

The pseudocode for T-KNOW is shown in Figure 3. *CN* is the total number of clues used. *clue(t,i)* is a function which returns a *query* string by concatenating the tag *t* with predefined clues (from 1 to *CN*). This query is searched on Google using the Google API. The function *download_google_abstracts(query,n)* takes the query and number of abstracts required as parameters and returns the abstracts of search results found for the given query. The cosine measure is calculated between each abstract (*a*) and context (*ctx*) of the tag (*t*). If the value of the cosine measure is above a certain threshold, then the abstract (*a*) is considered for further processing. Patterns (find the complete list in [3], and example in step 3) for clue *i* are matched against the abstract *a* using the function *pattern_match(a,i)*. If the pattern is matched, then the category of current tag is extracted. The category having the highest similarity with context of the tag is returned.

```
TKNOW(Tag t, Context ctx) {
  for i = 1 to CN {
    query = clue(t,i)
    abstracts = download_google_abstracts(query,n);
    foreach a in abstracts {
      sim = calculate_similarity(a,ctx);
      if (sim > threshold) {
        if (pattern_match(a,i)) {
          c = get_category(a);
          Res[c] = Res[c]+sim;
        }
      }
    }
  }
  return maxarg_c Res[c];
}
```

Fig. 3. Pseudocode of T-KNOW

3.1 Measuring similarity between search results and tags

There can be multiple ways for computing the similarity between the search result and the tag depending upon the context of the tag. We have proposed four methods (3.1.1 – 3.1.4) of selecting the context of the tag. For measuring similarity between Google search result and the context of a tag, the cosine measure is computed between the

bag of word representations of the abstract of the downloaded search result $\vec{a}$ and the context $\vec{C}$ of the tag *t*. If this cosine measure is above a certain threshold value, the result is considered for further processing. The cosine measure is calculated as

$$\cos(\angle(\vec{C},\vec{a})) = \frac{\vec{C}.\vec{a}}{\|\vec{C}\|.\|\vec{a}\|} \quad \textbf{(1)}$$

Section 4.2 presents different results obtained using different threshold values and different contexts. To understand the different contexts, consider the images in Figure 4. The left most image is of "Eiffel Tower". The middle image is "Notre Dame". The right most is the image of a Cow. Table 1 shows the details of each image, its tags, the user who has uploaded this image, and the group in which this image is present.

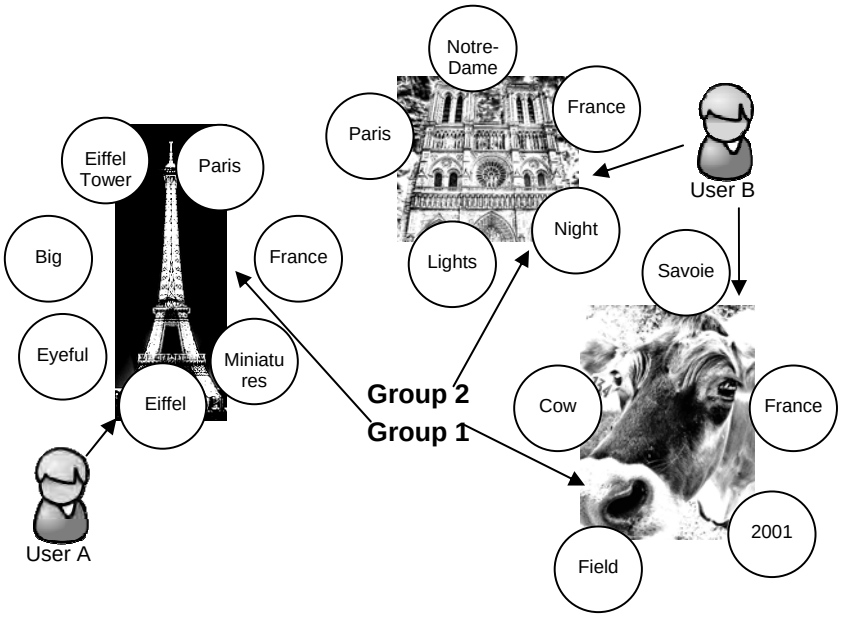


Fig. 4. Sample images with tags

Table 1. Details of images in Figure 4

Image	Tags	User	Group
Eiffel Tower	Eiffel Tower, Paris, France, Miniatures, Eiffel, Eyeful, Big	A	1
Notre Dame	Notre-Dame, France, Night, Lights, Paris	B	2
Cow	Savoie, France, 2001, Field, Cow	B	1

To formally define the context of the tags, we need to formally define the tagging

systems. We use the same formal model of tagging systems (or folksonomies) as defined in [10]. According to [10] a tagging system (or folksonomy) is a tuple

$$F := (U,T,R,Y) \quad (2)$$

where U, T, and R are finite sets representing users, tags, and resources respectively, Y represents taggings by users U, using tags T of resources R, and $Y \subseteq U \times T \times R$. In addition to these sets, we also use the set of groups G that might be found in some tagging systems (like Flickr). Users can post their resources to these groups. Now we formally define different contexts as

3.1.1 Resource Context (R)

In order to represent a tag by its context, we here consider the case of resource context. We choose the tags that belong to the current resource except the tag (to be classified) itself. The Resource Context of tag t for resource r can be defined as

$$C_R(t,r) = \left\{ t' \in T \setminus \{t\} \middle| (u,t',r) \in Y \wedge u \in U \right\} \quad (3)$$

We are also interested in the frequency of t_i in resource r_j (in case of Flickr it is at most 1, because one tag can occur only at most once in a resource) to create a bag of words using this context. $W_R(t,r)$ represents the number of times tag t appears with resource r.

$$W_R(t,r) = \left| \left\{ (u,t,r) \in Y \middle| u \in U \right\} \right| \quad (4)$$

We can get the Resource Context of a tag t of resource r using $C_R(t,r)$ and for each tag t_x in the Resource Context of tag t, we can get its number of occurrences in resource r_x using $W_{CL}(t_x,r_x)$.

We can define a bag-of-words resource context representation of a tag t appearing in resource r, i.e. by

$$B_R(t,r) = \left\{ \left(t', W_R(t',r) \right) \middle| t' \in C_R(t,r) \right\} \quad (5)$$

Note that B_T, B_{SU}, and B_{SG} can be defined in the similar manner for Tag, Social User, and Social Group contexts respectively. Consider that we want to classify the tag "Paris" of the image Eiffel-Tower in Figure 4, only the tags of the image Eiffel-Tower are selected as the context, i.e. C_R("Paris",Eiffel-Tower)={"Eiffel Tower", "France", "Miniatures", "Eiffel", "Eyeful", "Big"}. The bag-of-words representation of the tag "Paris" of Eiffel-Tower will be B_R("Paris",Eiffel-Tower)={("Eiffel Tower",1), ("France",1), ("Miniatures",1), ("Eiffel",1), ("Eyeful",1), ("Big",1)}.

3.1.2 Tag Context (T)

In case of Tag Context, we select all the tags joint to the resources having the tag t, except the tag t itself. Tag Context can be defined as

$$C_T(t) = \left\{ t' \in T \setminus \{t\} \middle| (u,t,r) \in Y \wedge (u',t',r) \in Y \wedge u \in U \wedge u' \in U \wedge r \in R \right\} \quad (6)$$

For creating a bag of words representation (like (5)) using this context, we define $W_T(t,t')$ that represents the number of times tag t appears with tag t'.

$$W_T(t,t') = \left| \left\{ (u',t',r) \in Y \middle| (u,t,r) \in Y \wedge u \in U \wedge u' \in U \wedge r \in R \right\} \right| \quad (7)$$

We can get the Tag Context of a tag t using $C_G(t)$ and for each tag t' in the Tag Context of tag t, we can get its number of occurrences with tag t using $W_T(t,t')$.

Consider that we want to classify the tag "Paris" of the image Eiffel-Tower. All tags of images having the tag "Paris" are selected as the Tag Context except the tag "Paris" itself. In example of Figure 4, Eiffel-Tower and Notre-Dame have the tag "Paris", so all the tags of the images Eiffel-Tower and Notre-Dame are added to the context of the tag "Paris" except the tag "Paris" itself, and number of occurrences of each of these tags with tag t can be calculated using W_T. Thus, B_T("Paris", Eiffel-Tower})={("Eiffel Tower",1), ("France",2), ("Miniatures",1), ("Eiffel",1), ("Eyeful",1), ("Big",1), ("Notre-Dame",1), ("Night",1), ("Lights",1)} is the bag-of-word representation constructed using Tag Context of the tag "Paris".

3.1.3 Social User Context (SU)

In case of Social User Context of a tag t, we select all the tags used by a user u, except the tag t itself. Social User Context of tag t of user u can be defined as

$$C_{SU}(t,u) = \left\{ t' \in T \setminus \{t\} \middle| (u,t',r) \in Y \wedge r \in R \right\} \quad (8)$$

For creating a bag-of-words representation (like (5)) using this context, we define $W_{SU}(t,u)$ that represents the number of times tag t is used be the user u.

$$W_{SU}(t,u) = \left| \left\{ (u,t,r) \in Y \middle| r \in R \right\} \right| \quad (9)$$

Consider that we want to classify the tag "Paris" of the image Notre-Dame that belongs to user B. All tags of images that belong to the user B are selected as the context except the tag "Paris" itself. In example of Figure 4, the images Notre-Dame and Cow belong to the user B, so all the tags of the images Notre-Dame and Cow are added to the context of the tag "Paris" except the tag "Paris". Thus, B_{SU}("Paris", Notre-Dame)={("Notre Dame",1), ("France",2), ("Night",1), ("Lights",1), ("Savoie",1), ("2001",1), ("Field",1), ("Cow",1)} is the bag-of-word representation constructed using social user context.

3.1.4 Social Group Context (SG)

In case of Social Group Context of tag t that is present in groups g, we select all the tags of all resources present in the same group g, except the tag t itself. The Social Group Context can be defined as

$$C_{SG}(t,g) = \left\{ t' \in T \setminus \{t\} \middle| (u,t',r) \in Y \wedge u \in U \wedge r \in R \wedge g \in Group(u,r) \right\} \quad (\mathbf{10})$$

where *Group(u,r)* is a function which returns the groups that contain the user *u* and resource *r*.

For creating a bag-of-words representation using this context (like in (5)), we define $W_{SG}(t,g)$ that represents the number of times tag *t* appears in the group *g*.

$$W_{SG}(t,g) = \left| \left\{ (u,t,r) \in Y \middle| u \in U \wedge r \in R \wedge g \in Group(u,r) \right\} \right| \quad (\mathbf{11})$$

Consider that we want to classify the tag "Paris" of the image Eiffel-Tower that belongs to group 1. All tags of images present in group 1 are selected as the context except the tag "Paris" itself. In example of Figure 4, the images Eiffel-Tower and Cow are present in group 1, so all the tags of the images Eiffel Tower and Cow are added to the context of the tag "Paris" except the tag "Paris" itself. B_{SG}("Paris",group-1})={("Eiffel Tower",1), ("France",2), ("Miniatures",1), ("Eiffel",1), ("Eyeful",1), ("Big",1), ("Savoie",1), ("2001",1), ("Field",1), ("Cow",1)} is the bag-of-word representation constructed using social group context.

4 Evaluation

In order to evaluate our system, we have used images, tags, user, and group information from Flickr website. We asked two persons to classify the data into four categories. We have then classified the same data set using T-KNOW in order to evaluate T-KNOW.

4.1 Experimental Setup

To organize tags into predefined categories, we have chosen four categories "Person", "Location", "Vehicle", and "Organization". To get ontologies related to these categories, we have searched Swoogle[1] [5] for general purpose ontologies and used the ontology OntoSem[2]. For this ontology, we have used concepts and sub-concepts of p1:vehicle, p1:organization, p1:place, p1:geopolitical-entity, and p1:human as categories. We have used a total of 932 concepts as categories out of this ontology.

After selecting the categories, we have gathered data from groups present at the Flickr website. Users can post their images to different groups on Flickr. One group usually contains images related to the topic of that group. For example, the vehicles group contains images of vehicles. We have searched for groups related to the topics (i) people, (ii) locations, and (iii) vehicles using the group search facility provided by Flickr, and then selected three groups from each topic. We have selected only those

[1] http://swoogle.umbc.edu/

[2] http://morpheus.cs.umbc.edu/aks1/ontosem.owl (last accessed Mach 21, 2007)

groups which had at least 100 images and 25 members. The groups selected were candid_celebrity, 35212032@N00 (famous people), politicians, CarDirectory, classic_cars, vehicles, PraiseAndCurseOfTheCity, signcity, and cities. Out of these groups, only the “famous people” group had 27 members and 165 images, all other groups had at least 100 members and more than 500 images. We have then randomly selected 21 images from each of these nine groups. There were a total of 1754 tags in all of these 189 images.

We asked two persons K and S (human classifiers) to classify the tags. They did not have any kind of information about this research and method. They have classified all the tags regardless of the language and spelling mistakes, which has of course affected the results of T-KNOW because T-KNOW uses English patterns for identifying categories. For example, the users have classified the tags “Russia” and “Russland” (German word for Russia) as location, whereas T-KNOW was unable to identify “Russland”, as this is not an English word and hence is not supported by the pattern library used. A spreadsheet was provided to each human classifier with resources, tags, and links to the original Flickr images, Wikipedia, and Google. For example if a user finds a tag "Essen" (a German city as well as the German word for meal) and is unable to decide about its category, he can view the image (in which this tag is present) on Flickr website, if this image is not helpful to identify the tag, he can search it in Wikipedia[1], and still if it unclear, then he can find it in Google[2]. Human classifiers (K and S) agreed upon classification of only 1166 tags out of 1754 tags.

4.2 Results

This section contains the results obtained by classifying tags using T-KNOW with different contexts and threshold values. Table 2 shows the number of tags and resources classified manually (by user K) and using T-KNOW with threshold of 0.0 and Social Group (SG) context.

Table 2. Number of tags and resources classified per category by User K and T-KNOW with Threshold = 0 and Social Group (SG) context

	Resources		**Tags**	
Category	**User K**	**th=0, SG**	**User K**	**th=0, SG**
Location	139	155	519	485
Organization	39	54	89	67
Person	86	107	287	229
Vehicle	69	64	259	109
Other	155	177	600	864

[1] http://en.wikipedia.org/wiki/Special:Search/essen

[2] http://www.google.com/search?num=100&hl=en&lr=&as_qdr=all&q="essen"&btnG=Search

We have used F-measure and Cohen's Kappa for evaluation of our method. F-measure is a common measure in information retrieval, in case of tags classification we have computed F-measure as, if

$$A = \textit{set of correct classification by test}$$

$$B = \textit{set of all classification by Gold Standard}$$

$$C = \textit{set of all classifications by test}$$

(In our evaluation, user K is the *gold standard*, and *test* is either user S or T-ORG)

then, we define Precision, Recall, and F-measure as

$$Precision = \frac{A}{C} \quad (\mathbf{12})$$

$$Recall = \frac{A}{B} \quad (\mathbf{13})$$

$$F\text{-}measure = \frac{2 * \mathrm{Pr}ecision * \mathrm{Re}call}{\mathrm{Pr}ecision + \mathrm{Re}call} \quad (\mathbf{14})$$

Figure 5 displays the F-measure with user K defining the gold standard and T-KNOW using different threshold values and contexts and it also shows the F-measure of the classification of user K and user S (shown as a constant line).

Due to the possibility of classification that might occur just by chance, we have also calculated the Cohen's Kappa [4] between a user's classification and the system's prediction. Cohen's Kappa is defined as

$$K = \frac{P_0 - P_c}{1 - P_c} \quad (\mathbf{15})$$

where P_0 is the observed agreement between classifiers and P_c is the agreement occurred due to chance. If the two classifiers agree completely, then the value of Cohen's Kappa is 1. Figure 6 shows the Kappa values of the classification of user K and T-KNOW (with different threshold values and contexts) and it also shows the Cohen's Kappa value between the classifications of user K and user S (shown as a straight line).

4.3 Discussion

The task of organizing resources by classifying tags in a tagging system is not trivial. It is observed that two humans classifying the same data set might not totally agree with each other, as observed in the case of humans classifiers of user K and user S, the kappa value was 0.53, whereas this value would be 1 in case of complete agreement between classifiers.

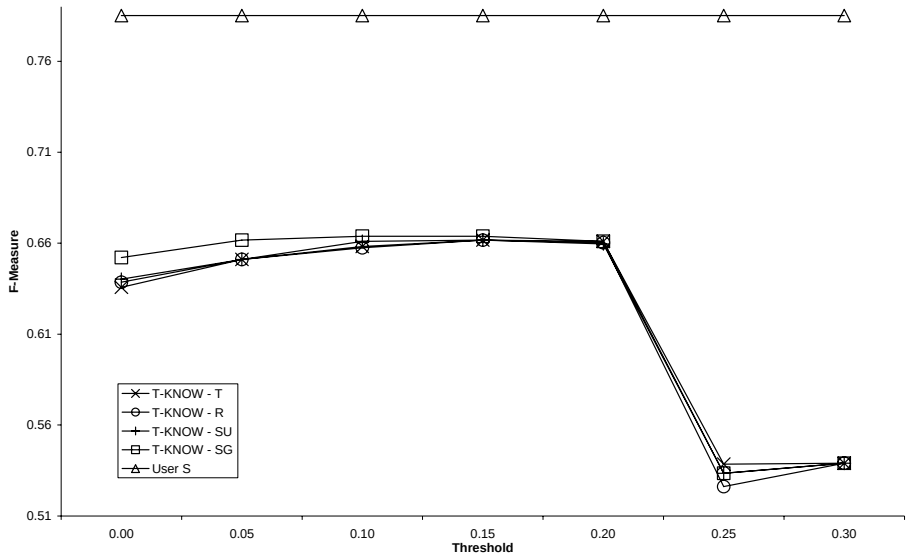


Fig. 5. F-Measure with user K defining the gold standard

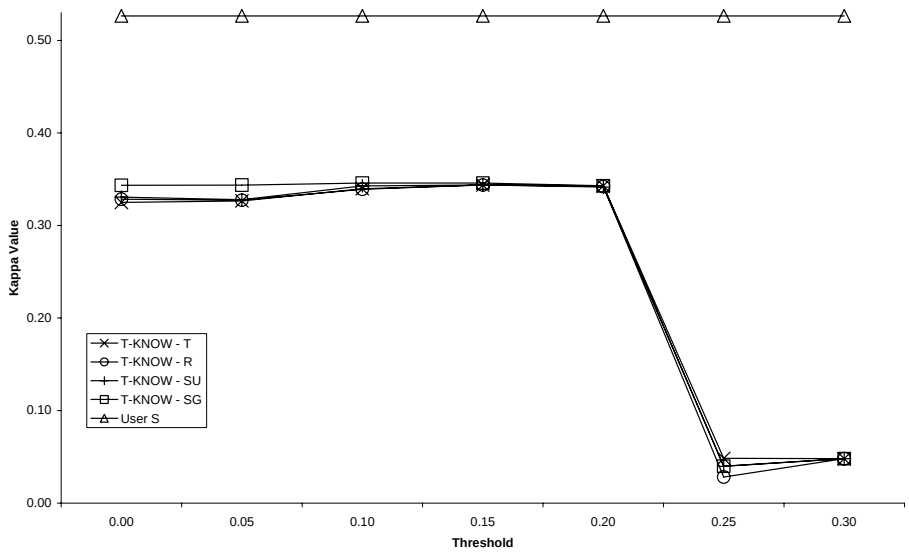


Fig. 6. Cohen's Kappa values for classification of T-KNOW and User S with user K defining the gold standard

Table 2 shows the number of tags and resources per category. The difference between number of resources or tags classified by different classifiers per category is small. As the average resources per user were 1.39 in the data set, the difference between F-measures of Resource (R) and Social User (SU) contexts is hardly visible. We believe

that if there are more resources per user, then the results of classification will be different for these context types. The best F-measure obtained was 0.66 with the context Social Group (SG) at thresholds of 0.10 and 0.15 and this small advantage was stable over other thresholds except 0.25. The F-measure is affected by the problem of classification by chance. Therefore we have calculated Cohen's Kappa [4] to measure the agreement between two users and between T-KNOW and user K. The majority class ("Other" in our case) scores zero in Cohen's Kappa [4]. F-measure lacks this property. The Cohen's Kappa between classification of users K and S was 0.53 (shown as a straight line in Figure 6), which shows the disagreement between the classifications of human users. Best kappa value for gold standard (user K) was 0.35 with Social Group (SG) context and using threshold of 0.10 or 0.15.

The results show that, the different approaches for selecting a context are statistically not significantly different. Keeping in view the small difference between different approaches, Social Group (SG) context has given overall better results as compared to other contexts. This is because the tags which are chosen as context belong to the same type of resources/images (as a group mostly contains same type of resources). In case of other contexts, tags of the resources with different subjects might be selected as context, which might affect the similarity measure.

5 Related Work

Tagging systems are becoming popular and more people are using them, especially in a social environment. A general overview of tagging systems can be found in [6]. Schmitz et al. have formalized folksonomies and discuss the induction of association rule mining for analyzing and structuring folksonomies in [10]. A lot of work has been done to extract useful information using natural language patterns. Hearst has used lexico-syntactic patterns to extract hyponyms from large text corpora [7]. Our approach is based on the matching of such lexico-syntactic patterns. These linguistic patterns have been used by other researchers for semantic annotation. T-KNOW is based on the C-PANKOW system (see [2] and [3]), which uses lexical patterns along with Google for semantic annotation of web pages. We have used and formally defined context of tags for measuring similarity between Google search results and the tags, contextual information has also been used by others, like [1] has used contextual information in recommendation process. [9] has used context information and Google search for identifying color of an object, which helps in clustering of images.

6 Conclusion

We have presented T-ORG to organize resources in a tagging system. T-ORG uses T-KNOW for unsupervised classification of tags which exploits Google and linguistic patterns. We have proposed four ways to context of a tag. Experimental results show that the classification accuracy for this unsupervised method is indeed encouraging,

especially in the light of the low agreement between the classifications done by two humans. Subsequent user experiments should show whether such classifications help to improve the user experience in a system like /facet [8].

Acknowledgments. We would like to acknowledge Higher Education Commission of Pakistan and German Academic Exchange Service (DAAD) for providing scholarship and support to Rabeeh Abbasi for conducting his PhD. This work has been partially supported by the European project "Semiotic Dynamics in Online Social Communities" (Tagora, FP6-2005-34721). We also thank Dr. (med.) Saadullah Abbasi and Ms. Katrin Michels for classifying the tags manually.

References

[1] Adomavicius, G., Sankaranarayanan, R., Sen, S., and Tuzhilin, A. 2005. Incorporating contextual information in recommender systems using a multidimensional approach. J-TOIS, 23(1), 103–145.

[2] Cimiano, P., Handschuh, S., and Staab, S. 2004. Towards the self-annotating web. In Proceedings of the 13th international Conference on World Wide Web (New York, NY, USA, May 17 - 20, 2004). WWW '04. ACM Press, New York, NY, 462-471.

[3] Cimiano, P., Ladwig, G., and Staab, S. 2005. Gimme' the context: context-driven automatic semantic annotation with C-PANKOW. In Proceedings of the 14th international Conference on World Wide Web (Chiba, Japan, May 10 - 14, 2005). WWW '05. ACM Press, New York, NY, 332-341.

[4] Cohen, J. 1960. A coefficient of agreement for nominal scales. Educational and Psychological Measurement, 20(1), 37-46

[5] Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R. S., Peng, Y., Reddivari, P., Doshi, V. C., and Sachs, J. 2004. Swoogle: A semantic web search and metadata engine. In Proc. 13th ACM Conf. on Information and Knowledge Management.

[6] Golder, S. and Huberman, A. B. 2006. The structure of collaborative tagging systems. Journal of Information Science, 32(2):198-208, April 2006.

[7] Hearst. M.A. 1992. Automatic acquisition of hyponyms from large text corpora. In Proceedings of the 14th International Conference on Computational Linguistics.

[8] Hildebrand, M., Ossenbruggen, J., Hardman, L. 2006. /facet: A Browser for Heterogeneous Semantic Web Repositories. International Semantic Web Conference, 272-285.

[9] Millet, C., Grefenstette, G., Bloch, I., Moellic, P.A., Hede, P. 2006. Automatically populating an image ontology and semantic color filtering. International Workshop Ontoimage'2006 Language Resources for Content-Based Image Retrieval, Genoa, Italy, pp 34-39.

[10] Schmitz, C., Hotho, A., Jaschke, R., and Stumme, G. 2006. Mining association rules in folksonomies. Proceedings of the IFCS 2006 Conference.

Making the Semantic Web a Reality through Active Semantic Spaces

Yihong Ding[1], Ying Ding[2], David W. Embley[1], Omair Shafiq[2], and Martin Hepp[2]

[1] Department of Computer Science, Brigham Young University, U.S.A.
{ding,embley}@cs.byu.edu
[2] Digital Enterprise Research Institute (DERI), University of Innsbruck, Austria
{ying.ding,omair.shafig,martin.hepp}@deri.org

Abstract. One reason for the slow acceptance of the Semantic Web is the lack of simple, straightforward, and attractive end-user showcases. Interest from end-users is an essential driving force of all web technologies. In this position paper we propose active semantic spaces (ASpaces) as a possible showcase. An ASpace is built on a combination of Semantic Web technologies, service agent technologies, and Web 2.0 technologies. Semantic Web technologies support producing machine-processable content on ASpaces. Service-agent technologies support proactive machine agents on ASpaces that communicate with both their human users and their peers. Web 2.0 technologies support friendly user interaction and the capability of dynamically collecting remote feedback from ASpace agents. By combining these technologies together, users can issue personal requests to ASpace agents, which then look for answers on other ASpaces. ASpace agents automatically blog results back to users as if they had come from remote human users. This showcase is an example of bridging the gap between the Semantic Web and Web 2.0.

1 Introduction

Despite its importance, designing and producing a good Semantic-Web showcase is not easy. To qualify, the showcase must include machine understanding, a nontrivial requirement. To enable machine understanding, we are likely to have to carefully integrate foundation technologies including ontologies, semantic annotation, and semantic search. At the same time, however, we must not make showcases too complex. Satisfying all these requirements is a huge challenge.

In this position paper, we present the idea of Active Semantic Spaces (ASpaces), a potential Semantic-Web showcase, and we explain how, despite the huge challenge, it can be realized. As Figure 1 shows, an ASpace is a combination of three technologies. (1) Semantic Web technologies in an ASpace provide ontology-specified semantics about the ASpace owner and about domains of interest to the owner. (2) Web 2.0 technologies in an ASpace support friendly user interaction and the capability of dynamically collecting remote feedback. (3) Service agent technologies in an ASpace allow machine agents to communicate with each other. Together these technologies support four advances over current web technologies. First, ASpaces provide a new active human communication model between web readers and writers: blog writers can actively find potential readers rather than simply waiting for responses. Second, ASpaces provide query-answering services beyond the capability of current search engines. Third, ASpaces can exist simultaneously with the current web, and the wide adoption of ASpaces may help actualize the dream of the Semantic Web. Fourth, ASpace design is an example of bridging the Semantic Web and Web 2.0.

Our vision of the Semantic Web is close to the vision of Semantic Web 2.0 as discussed by Breslin and Decker [2]. Both of us agree that Web 2.0 is not enough, and we need to add richer semantics into Web 2.0 publications to provide users greater facilities to manipulate web data. The difference is, however, that we emphasize more on the side of enhanced machine communications, while Breslin and Decker focuse more on the side of facilitating human communications. We believe that both sides are crucial to the realization of the next-generation web.

2 ASpaces: A Semantic Web Showcase

A successful showcase must be pragmatic, which means that users can immediately see its value. For example, when a homepage (a traditional web showcase) is created, its developers can display it properly on their own as well as on other computers with internet connections. When a blog (a Web 2.0 showcase) is created, its developer can start to view feedback from readers of the blog immediately. Similarly, when an ASpace is created, its developers must be able to actively find some likely potential readers or desired information.

To make this work and be pragmatic, an ASpace contains ontologies, annotated content, agents, services, and blogging capabilities.

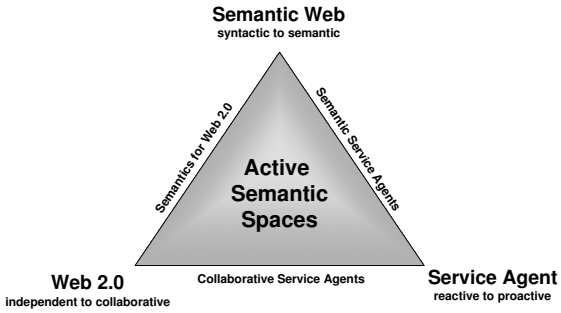


Fig. 1. Origin of Active Semantic Spaces.

2.1 ASpace Ontologies and Annotated Content

The need for ontologies is a major challenge for the Semantic Web. For ASpaces, we believe, as do others [6, 10], in two assumptions: (1) Semantic-Web users need only simple ontologies, and (2) personalization of ontologies simplifies mutual communication between humans and machines.

Based on these beliefs, we have adapted a light-weight ontology representation [4, 5] for ASpace ontologies. ASpace ontologies are equivalent to OWL-DL ontologies in formalism and reasoning power, but they differ from OWL because they include instance recognition semantics.[3] *Instance recognition semantics* are formal specifications that interpret instances of a concept in ordinary text. Syntactically, our declarations of instance recognition semantics allow users to specify recognition phrases, context phrases, and exception phrases in regular expressions for any ontology concept. For example, in a declaration of the concept *Product Price*, we can specify its recognition phrase to be "\d{1,4}(\.\d\d)?," which allows the range of prices to be from 0.00 to 9999.99. Optionally we can add a left-context phrase "\$"[4], context keywords "price" and "product" and an exception context keyword "discount." Therefore, this declaration can correctly recognize the number in "product price: $95.50" to be a legal instance of *Product Price*, and correctly exclude the number in "product discount: $5.00" as a *Product Price*.

Using ASpace ontologies, we can semi-automatically annotate web content. To help annotate web documents, we have developed an automated, ontology-based semantic annotation tool [4],[5] which ultimately is based on the ontology-

[3] We have proposed an extension to OWL that includes instance recognition semantics, and we have implemented conversions to and from OWL and our proprietary ontology language [3].

[4] Right-context phrases are also allowed though our example does not show it.

[5] Online demo is available at http://www.deg.byu.edu/.

based data-extraction technology we have studied and developed for years [5]. The core of this automated semantic annotation tool is the use of instance recognition semantics in ontologies. Our earlier experiments have shown that this semantic annotation tool can automatically annotate data-rich web content with high accuracy [4].

In addition to automated annotation, we also provide a simple manual annotation tool so that users can revise and update automatically created annotations. Users can also use the Web-2.0-style tagging techniques to categorize their documents with respect to selected ASpace ontologies. We store both automatically created and manually created annotated data in RDF files.

2.2 ASpace Blogs and Services

ASpaces are mediators that connect web users to the public web. Each ASpace contains two types of blogs: publication blogs (PuBs) and request blogs (ReBs). Through PuBs, users publish their information to the public. Through ReBs, users issue requests to collect information of interest from the public web. Both PuBs and ReBs are linked to a common local ontology repository, and to the way be annotated. This design is close to the idea of semantic blogs and semantic wikis. Karger and Quan [7], for example, view blogging as a user-friendly way to exchange data and encourage semantic annotation of blogs. Later on, Möller and his colleagues [8] developed a *semiBlog* editor to assist in adding metadata to blog posts. The annotations can be added to individual posts in RDF format through which machines may find connections between different blogs. These approaches, however, have not addressed personalized knowledge specifications and have not enabled users to issue personal requests for active web search.

In ASpaces, PuBs are standard blogs augmented with semantic annotation. Semantic annotations aid remote machine agents by giving them specific information about these PuBs with respect to an ontology. Without annotations, PuBs are assumed to carry only latent information because it is not guaranteed that remote agents can understand web content without annotations. Whether unannotated content can be understood depends on the ability of an agent to automatically annotate the content with respect to a known ontology. PuB owners have complete freedom to decide the percentage and in how much detail they want to annotate their PuBs. Certainly, if the percentage and detail of annotated content is greater, the chance of a PuB may become useful and used by remote agents is greater. This design strategy gives users freedom of choice as well as the motivation to do detailed annotations.

ReBs are private blogs (visible only to the owner) working as communication interfaces between ASpace users and ASpace agents. Users write requests on ReBs and invoke machine agents to understand and execute them. User requests are annotated, and thus aligned with ontologies, by the same automatic and manual annotation module used to annotate PuBs. Once annotated requests become machine-processable, they are converted to executable queries and executed on PuBs at remote ASpaces. After requests are executed, machines can

automatically blog the results back on ReBs as if they had come from remote human users.

An ASpace is not only a space that stores user information (PuBs), but also ASpace agents in which machines execute commands and interact with users (ReBs). This is why designing the space as being "active." Since the purpose of Semantic Web is to leverage machine processing, a Semantic Web showcase must contain machine agents. Human readers do not need "machine-processable" content.

We now step through a simple example to show how ASpaces behave and why ASpaces can be a Semantic Web showcase.

Story. Suppose Bob, an ASpace user, likes Nikon Coolpix S5 digital cameras. He wants to become acquainted with people who own or have an interest in this product. He also wants to find coupons for this product for both himself as well as anyone else who might be interested. This story, though simple, has a complicated scenario that cannot be resolved well on the current web. For example, finding people who own or have an interest in Nikon Coolpix S5 is very tedious. Searching for people based on their interests is not well supported on the current web. Searching by product name often results in hundreds of sales, manufacturer, and review pages prior to personal homepages (or blogs) that contain product information. Coupons for a specific product may or may not be easy to find, but even after Bob has found a coupon, it is not easy for Bob to appropriately notify others about this coupon. Since Bob does not really know if others he has found as a result of his search are also interested in these coupons, Bob should avoid broadcasting an email message, which may be received as bothersome spam. ASpaces address all the issues.

Ontologies. This story is about three small domain ontologies: person contact information, digital cameras, and digital coupons. We support initial ASpace creation by providing an array of ontologies including instance recognition semantics for common domains. Ontologies for contact information, digital cameras, and coupons would be among them.

Requests. After Bob has selected ontologies of interest, he can start writing his requests on his ReB. First he writes "Find people who own or have an interest in Nikon Coolpix S5 digital camera." The annotation module in Bob's ASpace annotates this request as follows.

"Find <person-contact-info:Person>people</...> who own
or have an interest in <digital-camera:Make>Nikon</...>
<digital-camera:Model>Coolpix S5</...>
<digital-camera:Digital Camera>digital camera</...>."

Next, Bob writes "Search for coupons for Nikon Coolpix S5 cameras that remain valid until the end of this month." His annotation module annotates this request as follows.

> "Search for <digital-coupon:Coupon>coupons</...> for
> <digital-coupon:Product Name>Nikon Coolpix S5</...> cameras that
> remain valid until <digital-coupon:Valid Time>the end of this month</...>."

After Bob has done on requesting, Bob decides that it is the golfing time. So he simply leaves his ASpace agent take care of the rest of the tasks.

Request Execution. Bob's ASpace agent can perform the two requests simultaneously. For the first request, Bob's agent first contacts the ASpace servers to obtain a list of ASpaces that have also downloaded both of the person-contact-information and digital-camera ontologies. Bob's agent enumerates this list and contacts respective remote ASpace agents individually to request checking the annotated content on their PuBs. Once the checking request is granted, it matches the annotated content in the request and in the remote PuB based on common concepts. For example, it checks whether in the remote PuB the annotated *digital-camera:Make* is "Nikon." Once there is a match, it automatically blogs all the annotated record as well as the URL of the respective remote ASpace back to its own ReB. For example, if an agent find a match of "Nikon" and "Coolpix S5" as *Make* and *Model* of digital cameras on Alice's ASpace, it automatically blogs the annotated personal contact information of Alice (*person-contact-info:Person*) to its ReB as well as a link to Alice's ASpace. Execution of the second request is similar.

Wrap Up. Bob has found many new acquaintances who share a common interest and has also some valuable coupons. Bob wants to share coupons with his acquaintances, but he does not want to produce what may perceive to be spam. First Bob clicks on a button to tell his agent to transfer his newly acquired coupon information on the ReB to his PuB. Bob then asks his ASpace agent to send a notice to the agents of these new acquaintances. The notice tells agents at its target that this site knows about coupons for "Nikon Coolpix S5" cameras valid to the end of this month. The notice is stored latently on the remote ASpaces so that their owners would not be annoyed about reading unexpected information. If advocated by their owners, ASpace agents can directly go to Bob's site to get the information rather than first having to contact ASpace servers.

3 Evaluation and Discussion

ASpaces are a potential, attractive showcase for the Semantic Web. ASpaces allow web users publishing everything they can now publish. As an added value, however, ASpaces allow web users to issue many requests that cannot currently be resolved. People have spent too much time digging a little piece of useful information out of tons of uninteresting online publications. By letting ASpace agents execute human requests, people can gain time back and do what they really enjoy.

Our proposed ASpace implementation is pragmatic and flexible. Most of the new core technologies required to build the key components of ASpaces have already been developed: ontologies with instance recognition semantics [5], ontology-based automated semantic annotation [4], automated query generation from annotated user requests [1, 4], and agent communication in the Semantic Web [9]. Although we have decided to use these technologies in our implementation, our ASpace implementation is also open to other technologies. For example, annotation can be presented in either RDF or Microformat and request queries can be formatted with SPARQL or XQuery. As long as they share the same ontologies, different implementations of ASpaces would not prohibit communication among ASpaces.

For end users, setting up ASpaces is the same as setting up blogs, except that now they set up two different types of blogs: PuBs and ReBs. They also choose ontologies from an ASpace server. This is, however, similar to do online bookmarking using del.icio.us, which is now a quite popular and an accepted online human activity. At this point the ASpace is set up, but not as useful as it could be. To make them much more useful, ASpace users can invoke downloaded extraction ontologies to automatically annotate existing blogs and web pages. To make annotated blogs and web pages even more useful, ASpace users can manually correct and add annotations.

What is not straightforward is to construct the ASpace server library of ontologies. Experience shows that it takes a few dozen person hours to construct an ontology for a domain like digital cameras. To aid in this process, we have begun to work on tools to semi-automatically construct ontologies.[6]

4 Concluding Remarks

We emphasize that the major contribution of this ASpace project to the community of the World Wide Web is not its implementation, though this implementation is also a good contribution. It is its design, its philosophy, and its potential impact to the evolution of the web. ASpace is a novel example of bridging the gap among the three parties: from syntactic data display to semantic data description (Semantic Web), from independent web behaviors to collaborative web behaviors (Web 2.0), and from a reactive system to a proactive system (service agent).

5 Acknowledgement

Yihong Ding and David W. Embley are supported in part by NSF grant #0414644. Part of the work presented in this paper is also supported by the European Commission under the projects DIP (FP6-507483), SUPER (FP6-026850), and MUSING (FP6-027097), by the Austrian BMVIT/FFG under the FIT-IT project myOntology (grant no. 812515/9284), and by a Young Researcher's Grant from the University of Innsbruck.

[6] TANGO: Table ANalysis for Generating Ontologies, NSF grant #0414644

References

1. M. Al-Muhammed and D. W. Embley. Ontology-based constraint recognition for free-form service requests. In *Proceedings of 23th IEEE International Conference on Data Engineering (ICDE 2007)*, Istanbul, Turkey, April 2007. (in press).
2. J. Breslin and S. Decker. Semantic Web 2.0: Creating Social Semantic Information Spaces. In *Tutorial in the 15th International World Wide Web Conference (WWW 2006)*, Edinburgh, Scotland, May 2006.
3. Y. Ding, D. Embley, and S. Liddle. Building a bridge between ontologies and automated semantic annotation with OWL-AA. (submitted for review).
4. Y. Ding, D. Embley, and S. Liddle. Automatic creation and simplified querying of semantic web content: An approach based on information-extraction ontologies. In *Proceedings of the First Asian Semantic Web Conference (ASWC 2006) LNCS 4185*, pages 400–414, Beijing, China, September 2006.
5. D. Embley, D. Campbell, Y. Jiang, S. Liddle, D. Lonsdale, Y.-K. Ng, and R. Smith. Conceptual-model-based data extraction from multiple-record web pages. *Data & Knowledge Engineering*, 31(3):227–251, November 1999.
6. A. Iskold. The road to the Semantic Web. http://www.readwriteweb.com-/archives/the_road_to_the.php.
7. D. Karger and D. Quan. What would it mean to blog on the Semantic Web? In *Proceedings of the Third International Semantic Web Conference (ISWC 2004)*, pages 214–228, Hiroshima, Japan, November 2004.
8. K. Möller, J. Breslin, and S. Decker. semiblog—semantic publishing of desktop data. In *Proceedings of the 14th Conference on Information Systems Development (ISD 2005)*, Karlstad, Sweden, August 2005.
9. O. Shafiq, Y. Ding, and D. Fensel. Enabling interoperability between information systems by bridging multi agent systems and web services. In *Proceedings of 10th IEEE International Enterprise Computing Conference (EDOC 2006)*, Hong Kong, China, October 2006.
10. N. Spivack. What is the Semantic Web, actually? http://novaspivack.typepad.-com/nova_spivack_weblog/2006/11/what_is_the_sem.html.

Towards Scientific Collaboration in a Semantic Wiki

Christoph Lange

Computer Science, Jacobs University Bremen*, ch.lange@jacobs-university.de

Abstract. On the Web 2.0, there are numerous projects for collaboratively creating and using scientific knowledge in a wiki—think of the scientific sections of *Wikipedia* or domain-specific platforms like *PlanetMath*. They do, however, not yet offer semantic services that could promote collaboration both of scientific knowledge engineers and of scholars or that take semantics emerged from such communities or acquired from page contents into account.
On the other hand, there are several semantic wikis—wikis enhanced with Semantic Web technologies. Current semantic wikis, however, only offer rather generic semantic services, such as semantic navigation, semantic-based editing assistance, and semantic search. Semantic services tailored to scientific knowledge and its specific structures (e. g. theories depending upon each other) are not yet provided.
Based on the argument that current semantic wikis lack scientific services because domain-specific ontologies are not properly integrated, this article proposes the basic architecture of a semantic wiki centered around an ontology of scientific markup languages. Two services to be designed on top of this ontology abstraction layer are outlined, and suggestions on how to improve them by making them community-aware are discussed.

1 State of the Art and Problem Statement

Current wiki projects for scientific knowledge range from comprehensive encyclopediae like *Wikipedia*, which covers all domains, even non-scientific ones, to projects specialised to a particular domain, such as *PlanetMath*, a wiki for mathematics[1]. As new content can quickly and easily be created and linked, wikis are also suitable for corporate knowledge management [9]—and for teams of scientists in a similar way. Neither *Wikipedia* nor *PlanetMath* offer certain services desirable for scientific communities, though.

A non-semantic wiki lacks a deeper understanding of the network of links between its pages. Semantic wikis [20] address this problem by typing pages and links with terms from ontologies [14]; usually, one page describes one real-world concept (e. g. a scientific theory). Page and link types can serve as the foundation for semantic services. Two services that are desirable in a scientific

* formerly International University Bremen

[1] See `http://www.wikipedia.org` and `http://www.planetmath.org`, respectively.

community will be discussed in section 4: one that suggests topics to learners, and another one that manages dependencies among concepts, which is useful for theories in development. Solutions for both of these problems have been available on the Semantic Web for years (see [2] or [10], resp.), but not yet integrated with (semantic) wikis on a large scale. A wiki extension for learning has been proposed with the *WikiTrails* system [16], which augments wiki content with navigation trails. These trails are either generated automatically by tracking user interaction, or they can be created manually (e. g. by a teacher), but the *knowledge contained in the wiki pages* is used in neither case. Integrating services that exploit this knowledge has been hampered by the fact that domain-specific ontologies are considered optional at best in most semantic wikis: They usually allow for ad-hoc modeling new ontologies or importing available ones [21], but as there is no uniform ontology layer at the *core* of these wikis, they cannot exploit characteristic traits of domain-specific knowledge.

Structural semantic markup is a common way to represent scientific knowledge. Available markup languages include OMDoc, a mathematical markup language that comprises and extends Content MathML and OpenMath [5], which only allow for representing formulæ, PhysML [1], an OMDoc variant adapted to physics, and the Chemical Markup Language CML[2] for chemical concepts like molecules and reactions. Semantic services for mathematical knowledge are the most advanced ones so far; for OMDoc, for example, services for learning assistance, semantic search, publishing (including community-specific notations of mathematical symbols), theory management, as well as proof verification have been developed [5, chapter 26]. Our work group, in collaboration with experts from scientific domains other than mathematics, is currently concerned with designing a unified "scientific markup language" and transferring these techologies to other domains, including physics, chemistry, geosciences, environmental sciences, and software engineering.

2 SWiM, a Semantic Wiki Prototype for Mathematics

Semantic wikis are appropriate for building "community-authored knowledge models" where informal natural language descriptions created by domain experts are formalised in collaboration with knowledge engineers [17]; the stepwise refining process of formalising human-readable texts they support is a common task for scientists [5]. Before scientific services can be implemented in a wiki, a base system supporting scientific documents must exist. So far, I have developed the SWiM prototype, a semantic wiki for mathematics [7], which is a modified *IkeWiki* [17] with OMDoc as its page format. It is capable of rendering OMDoc in a human-readable way using XSL transformations and extracting RDF triples used as typed navigation links from the markup. Other markup languages are not supported, and further semantic services are not yet offered. SWiM will serve as the base for implementing a wiki with services for scientific communities, tentatively named SWiM$^+$.

[2] `http://cml.sourceforge.net/`

As of March 2007, there is only one more semantic wiki dedicated to mathematics: se(ma)^{2}wi [23] is an experiment with a *Semantic MediaWiki* [19] fed with OMDoc-formatted mathematical knowledge from the *ActiveMath* learning environment [11]. Most of the structural semantics explicitly given in OMDoc is, however, lost during this import: The formulæ are converted to presentational-only LaTeX, and the links between wiki pages that represent mathematical statements, for example a link from a theorem to its proof, are not typed.

3 Representing Scientific Knowledge

To represent scientific knowledge, I follow the three-layered structure model of mathematics and science that M. Kohlhase has successfully applied to mathematics with OMDoc (see section 1): *Objects* (symbols, numbers, equations, molecules, etc.), *statements* (axioms, hypotheses, measurement results, examples, with relations like "proves", "defines", or "explains") and *theories*—collections of interrelated statements, which set symbols into their context[3]. This model has already been extended towards physics (PhysML) with just a few additions [1], and the PhysML creators anticipate that it also holds for other sciences.

For use in Semantic Web software, this model of scientific knowledge needs to be formally, explicitly specified in an *ontology*. The ontology behind the OMDoc markup format defines which knowledge can be represented in OMDoc and thereby approximates the general way of knowledge representation in mathematics. For the SWiM prototype, a subset of that ontology, which is given in a merely human-readable way in [5], has been explicitly modeled in OWL-DL: most statement types and their interrelations, theories and their "import" relation, with the addition of a generic transitive dependency relation. To make SWiM$^+$ support multiple scientific domains, ontologies of multiple markup languages will have to be formalised. Building on the work of the researchers working on a unification of scientific markup languages (cf. section 1), who will identify common traits of knowledge in all sciences covered—most likely including the three-layer stack of objects/statements/theories as well as generic containment and dependency relations—, one generic ontology, to be called "*upper* document ontology"[4] here, will be formalised in an appropriate language, such as OWL-DL[5].

One SWiM$^+$ page will most likely hold one statement or one small theory, which leads to small pages that are suitable for reuse by linking. As Semantic Web tools are not ready to use knowledge represented as markup in documents, relevant parts of it must first be *extracted* to a more formal representation like

[3] e. g. the glyph h as the height of a triangle in a theory of elementary geometry or Planck's quantum of action in a theory from quantum mechanics.

[4] A variation on the term "upper ontology", which the IEEE Standard Upper Ontology Working Group defines as an ontology "limited to concepts that are meta, generic, abstract and philosophical, and therefore are general enough to address (at a high level) a broad range of domain areas"; see `http://suo.ieee.org/`.

[5] A more formal definition of generic document ontologies is currently being developed by N. Müller and A. Mahnke, members of our group.

RDF (cf. [13]), using terms from the ontology behind the respective markup language. For example, a mathematical proof, marked up in OMDoc as `<proof xml:id="py-proof" for="pythagoras">`, would be represented by the two RDF triples `<py-proof, rdf:Type, om:Proof>` and `<py-proof, om:proves, pythagoras>`, terms from OMDoc's ontology being prefixed with `om:`. To make this extraction scale to multiple markup languages, solutions that use mappings from XML schemata to ontologies and back, such as *WEESA* [15], will be evaluated.

4 Semantic Services for Collaboration

Two key services envisaged for implementation within SWiM$^+$ are an interactive learning assistant for scholars and a dependency management assistant for scientists. In a work environment where scientists collaboratively formalise their ideas into theories, dependency management is an important feature: For example, if scientist E. In. Stein decides to base his newest ideas about "relativity" on assumptions about "gravitation" currently being developed by his colleague N. Ew. Ton and then Ton *changes* one of them, Stein's considerations might become invalid.

In an educational semantic wiki, course modules would be connected by links typed as "prerequisite". In terms of scientific markup, course modules can be realised as theories whose transitive "import" relation is interpreted as "prerequisite". If, for example, one member of a community interested in mathematics and its applications knows that, to fully understand MP3 encoding (let this be covered on a wiki page named *MP3Encoding*), one must know what a discrete cosine transform (DCT) is, he may connect these two pages with a "prerequisite" link. Imagine a second user who knows that orthogonal matrices are one basis for discrete cosine transforms and connect the pages DCT and *OrthogonalMatrix* accordingly. A Semantic Web reasoner can densify the network of knowledge by inferring additional knowledge not explicitly contained in the wiki pages, namely that reading (and understanding!) the wiki page *OrthogonalMatrix* is a prerequisite for fully understanding *MP3Encoding*. On the user interface, the direct and indirect prerequisites could then be recommended for reading.

The two services introduced here only rely on generic relations like dependency; therefore, they can be implemented on top of the above-mentioned "upper document ontology" and thus work across scientific domains. The same holds for two projected services that facilitate editing—ontology-based auto-completion of link targets and section-wise editing [8], but not for all envisioned services: Integrating the OMDoc-based formula search engine *MathWebSearch* [6], for example, is specific to the the domain of mathematics and requires access to full structural markup of formulæ instead of just extracted RDF triples.

5 Added Value from and for the Community

To design and improve services for SWiM$^+$ in a user-centered way, the method of "added-value analysis" [3] will be employed: First, specify a core problem, propose a solution for it, and establish the benefits and sacrifices of the solution, as perceived from a user's "micro-perspective". Benefits and sacrifices given, evaluate the core solution with regard to the core problem. Both benefits and sacrifices may spawn new core problems and thus ideas for further services—services that provide *added value* to the user [3].

5.1 Learning from the Community

Applied to the problem of (1) helping a user, named S. Ch. Olar here, to understand topics, an added-value analysis could lead to the following results: First we might propose the display of direct links from the current page on a navigation bar, grouped by types like "prerequisite" or "example",—a service offered by most semantic wikis, including *IkeWiki*, for free. The benefit of that solution is a concise overview of direct prerequisites and examples, at the expense of other direct links (e. g. from a topic to its type) also being shown and indirect prerequisites not being accessible. Taking up the latter sacrifice, we arrive at the new problem of (2) exploring direct *and* indirect prerequisites, which can be solved by computing all prerequisites beforehand and displaying links to all of them. The benefit is that all of them are now accessible within one click, at the expense that the list may contain too many links irrelevant to our user Olar.

Now, Olar needs to (3) distinguish relevant from irrelevant prerequisites—a ranking or pre-selection would be helpful. The social way, one could record how many of the prerequisites offered *other* readers of the same page actually clicked and rank or restrict the generated list based on that information. The benefit is that prerequisites most users considered irrelevant will not be included in the pre-selection computed for Olar and hence not distract him. A severe sacrifice is that this solution does not satisfy Olar's needs if they greatly differ from the needs of those who visited the respective page before. The new problem is to (4) give a better estimate of which preferences the user really has. In a single-user context, a *user model* containing the user's previous knowledge (e. g. obtained from his history of interaction with the system) and a user profile containing his learning goal and other preferences, such as notational ones, as in *ActiveMath* [11] would solve this problem.

A community-powered environment allows for giving improved estimates, though: If we assume that many users, divided into different sub-communities, are collaborating on one SWiM$^+$ site, problem (4) can be solved by finding out to which sub-community the user belongs. The lectora project [12] for enhancing the community-awareness of collaborative mathematical software, ran in our group, aims at improving search rankings and offering other services based on community models. While lectora is conceptually an independent system, the communication interface to SWiM$^+$ is currently being designed

in close cooperation[6]. LECTORA, connected to SWiM$^+$, would steadily be fed with information about all users' actions (reading, writing, annotating, setting preferences, ...). If LECTORA then finds out that OLAR belongs to the same subcommunity as another user, named L. E. ARNER, she henceforth recommends documents ARNER has rated as relevant or useful to OLAR—or, getting back to our added-value analysis, use this rating to rank a list of prerequisites computed by SWiM$^+$ for OLAR.

5.2 Better Collaboration for the Community

An added-value analysis starting from the problem of (1) managing dependencies between theories or course modules across changes could first lead to the following simple solution—applied to the two physicists from section 4: If R ("relativity") depends on G ("gravitation") and the user TON has changed G, thereby maybe breaking the current version of R, the next user to edit R—here: STEIN—could be notified that a page depended upon has been changed. This situation-dependent notification is a benefit over the usual list of recent changes in a wiki, which does not consider dependencies at all and which STEIN would have to visit on his own. Still, he has to make sacrifices: He needs to figure out whether TON has changed the semantics of G—instead of just fixing a typo, for example, —, and if so, whether that affects the consistency of R. If R is affected, STEIN must first figure out and then apply the appropriate change he has to make to R on his own. The problem that STEIN (2) does not know whether TON has changed the semantics of G could be solved the wiki way: Many wikis would allow TON to mark his change as "major" or "minor" [22]. This distinction is entirely subjective and thus not helpful, but we could offer the alternatives "semantics changed/not changed" instead. STEIN would benefit from that additional knowledge, but TON would have to make the sacrifice to correctly classify his change to G.

locutor [13], an ontology-driven management of change system developed in our group, will be a possible solution to the problem (3) of finding out whether one change actually affects other documents by computing "long-range effects" of changes. With its more detailed "taxonomy of change relations" it will provide a better solution to problem (2), too. If R is affected by the change of G, *locutor* will either be able to automatically make the required adaptations to R to restore consistency, or it will at least be able to pinpoint all effects of changes, to that STEIN would exactly know what to fix manually[7]. *locutor* is being implemented as an extension of the version management system *Subversion*[8]; thus, it can be integrated into SWiM$^+$ as a backend, replacing part of the SQL database used by the SWiM prototype.

[6] personal communication with CH. MÜLLER
[7] personal communication with N. MÜLLER
[8] See `http://kwarc.info/projects/locutor/` for a prototype.

6 Conclusion and Outlook

Mission ... Among others, the two services introduced in this article will be implemented in SWiM$^+$ and evaluated in long-term case studies in 2008, from which I expect feedback leading to further improvements. A scientific case study, focusing on dependency management and other services that support scientists in developing new knowledge, will be conducted in the cross-domain setting of the unified scientific markup language developed in collaboration with our group. An educational case study, focusing on the prerequisite learning assistant and on search facilities not presented here, will be conducted in an introductory course to computer science, the lecture notes of which are available in sTEX [4], an OMDoc-related format.

... and Vision: SWiM$^+$ will demonstrate how services on top of a semantic social software can make users benefit both from semantics extracted from formal documents and from semantics that emerged from communities of users. Following an idea from S. Schaffert [17], the achievements made in the "testbed" of a SWiM$^+$-driven site may then, thanks to the ontology abstraction layer, be transferred to the "large" web.

Acknowledgments The author would like to thank his colleagues Christine Müller and Normen Müller, as well as his advisor Michael Kohlhase, for productive discussions and for supporting his work, and the anonymous reviewers for broadening his horizon with their valuable suggestions.

References

1. E. Hilf, M. Kohlhase, and H. Stamerjohanns. Capturing the content of physics: Systems, observables, and experiments. In J. Borwein and W. M. Farmer, editors, *Mathematical Knowledge Management 2006*, number 4108 in Lecture Notes in Artificial Intelligence. Springer, 2006.
2. M. C. A. Klein, D. Fensel, A. Kiryakov, and D. Ognyanov. Ontology versioning and change detection on the web. In A. Gómez-Pérez and V. R. Benjamins, editors, *EKAW*, number 2473 in Lecture Notes in Computer Science, pages 197–212. Springer, 2002.
3. A. Kohlhase and N. Müller. Added-Value: Getting People into Semantic Work Environments. In *Emerging Technologies for Semantic Work Environments: Techniques, Methods, and Applications*. Idea Group, 2007. To appear; chapters under review.
4. M. Kohlhase. sTEX: A LaTeX-based workflow for OMDoc. In OMDoc [5], chapter 26.15.
5. M. Kohlhase. OMDoc – *An open markup format for mathematical documents [Version 1.2]*. Number 4180 in Lecture Notes in Artificial Intelligence. Springer Verlag, 2006.
6. M. Kohlhase and I. Şucan. A search engine for mathematical formulae. In T. Ida, J. Calmet, and D. Wang, editors, *Proceedings of Artificial Intelligence and Symbolic Computation (AISC)*, number 4120 in LNAI, pages 241–253. Springer Verlag, 2006.

7. C. Lange. SWiM – a semantic wiki for mathematical knowledge management. Technical Report 5, Jacobs University Bremen, 2007. http://kwarc.info/projects/swim/pubs/tr-swim.pdf.
8. C. Lange. Towards a Semantic Wiki for Science. http://kwarc.info/projects/swim/pubs/swimplus-resprop.pdf, 2007. Research proposal for a Ph. D. thesis.
9. B. Leuf and W. Cunningham. *The Wiki Way: Collaboration and Sharing on the Internet*. Addison-Wesley Professional, 2001.
10. M. d. O. Lidia Silva Muñoz, José Palazzo. Applying semantic web technologies to improve personalization and achieve interoperability between educational adaptive hypermedia systems. In *Proceedings of the International Workshop on Applications of Semantic Web Technologies for E-Learning (SW-EL)*, 2004.
11. E. Melis, G. Goguadze, A. G. Palomo, A. Frischauf, M. Homik, P. Libbrecht, and C. Ullrich. OMDoc in ActiveMath. In OMDoc [5], chapter 26.8.
12. C. Müller. Lectora – towards an interactive, collaborative reader for mathematical documents. http://kwarc.info/cmueller/papers/Mueller_ResearchProposal_2007-03-14.pdf, 2007. Research proposal for a Ph. D. thesis.
13. N. Müller. An Ontology-Driven Management of Change. In *LWA (Lernen, Wissensentdeckung und Adaptivität) conference proceedings*, 2006.
14. E. Oren, R. Delbru, K. Möller, M. Völkel, and S. Handschuh. Annotation and navigation in semantic wikis. In Völkel et al. [20].
15. G. Reif, H. Gall, and M. Jazayeri. WEESA: Web engineering for semantic web applications. In A. Ellis and T. Hagino, editors, *Proc. of the 14th WWW conference*, pages 722–729. ACM, 2005.
16. S. Reinhold. WikiTrails: Augmenting wiki structure for collaborative, interdisciplinary learning. In D. Riehle and J. Noble, editors, *Proceedings of the 2006 International Symposium on Wikis*, ACM Press, Aug. 2006.
17. S. Schaffert. Semantic social software – semantically enabled social software or socially enabled semantic web? In Sure and Schaffert [18].
18. Y. Sure and S. Schaffert, editors. *Semantics 2006: From Visions to Applications*, 2006.
19. M. Völkel, M. Krötzsch, D. Vrandečić, H. Haller, and R. Studer. Semantic Wikipedia. In *Proc. of the 15th WWW conference*, 2006.
20. M. Völkel, S. Schaffert, and S. Decker, editors. *1st Workshop on Semantic Wikis*, volume 206 of *CEUR Workshop Proceedings*, Budva, Montenegro, 2006.
21. D. Vrandečić and M. Krötzsch. Reusing ontological background knowledge in semantic wikis. In Völkel et al. [20].
22. Minor edit (from Wikimedia meta-wiki). http://meta.wikimedia.org/w/index.php?title=Help:Minor_edit&oldid=398318, 2006.
23. C. Zinn. Bootstrapping a semantic wiki application for learning mathematics. In Sure and Schaffert [18].

myOntology: The Marriage of Ontology Engineering and Collective Intelligence

Katharina Siorpaes and Martin Hepp

Digital Enterprise Research Institute (DERI), University of Innsbruck, Austria
katharina.siorpaes@deri.org, mhepp@computer.org

Abstract. Despite very active research on ontologies, only few useful ontologies can be found on the Web. The reasons for this are manifold, but a major obstacle is that ontology engineering environments impose high entrance barriers on users, and that the community does not have control over the ontology evolution. Wikis are a way to allow a wide range of users to contribute to Web representations without requiring more than basic Web editing skills. In the myOntology project, we propose the use of wiki technology in order to enable collaborative and community-driven ontology building by giving users with no or little expertise in ontology engineering the opportunity to contribute. In this paper, we describe the myOntology project in which the challenges of collaborative, community-driven, and wiki-based ontology engineering are investigated. Our approach combines the simplicity of wikis with intuitive visualization techniques and small yet efficient helper functionality plus consensus finding support exploiting the collective intelligence of a community.

Keywords: Ontologies, ontology engineering, collaborative ontology engineering, open ontologies, wikis

1 Introduction

Despite the active research on ontologies, only few useful ontologies can be found on the Web. The reasons for this are manifold including that ontology engineering environments impose high entrance barriers on users and the community does not have control over the ontology evolution. Wikis are a way to allow a wide range of users to contribute without requiring more than basic Web editing skills. In this paper we describe the myOntology project in which we use wiki technology in order to enable collaborative and community-driven ontology building by including users who have no or little expertise in ontology engineering. The remainder of this paper is organized as follows: in section 1, we outline the problem and motivate the paper. In section 1.3, we relate myOntology to previous works. In section 2, we describe the design principles and the architecture of our approach. In section 3, we sketch the implementation of the system. In section 4, we give a preliminary evaluation of our

approach by comparing traditional ontology engineering to the myOntology vision. In section 5, we summarize our findings.

1.1 Motivation

Ontologies are widely regarded as the "backbone of the Semantic Web" [1], [2] and the intensity of research on ontologies and related topics is very substantial - which can easily be shown by searching for the terms "ontology" and "ontologies" on Google Scholar[1], yielding 370.000 respectively 133.000 scholarly documents or references. However, when searching the Web, only few mature, practically useful ontologies can be found. This phenomenon has been discussed e.g. in [3], in which four bottlenecks were identified: First, many relevant domains of discourse, such as in e-commerce, show a high degree of conceptual dynamics, i.e. it is hard to keep up with the pace of change in reality. Second, the costs and potential benefits of building and using ontologies may be unfairly distributed among actors. Third, a prerequisite for using an ontology and thus committing to its view of the world is to be able to understand the meaning of concepts and relations. This is problematic for many users, since they cannot easily figure out what they would be committing to when using a particular ontology file from the Web. Fourth, when reusing existing consensus, e.g. in the form of standards or encyclopedias, one faces intellectual property rights, which means that ontologizing such input will require legal agreements with the current owners.

Currently, the most popular approach towards ontology building is engineering-oriented: a small group of engineers carefully builds and maintains a representation of their view of the world. However, a community-oriented approach where multiple individuals work on an ontology collaboratively has several advantages over an isolated engineering-oriented approach:

1. A community can keep up with the pace of conceptual dynamics in a domain more easily. Users have an interest in keeping the ontology up-to-date and therefore have a strong motivation to contribute to the maintenance.
2. The burden of creating the ontology can be distributed more evenly across those benefiting from the ontology.
3. The user community is more likely to agree on a view of the world that is represented by the ontology. Therefore, it is likely that this community will also actually use and further develop the ontology as it is not a subjective conceptualization based on a outdated state of the world.

However, we are currently lacking tool support that is suitable for ontology construction by large groups of non-experts over the Web. On the other hand, the philosophy of wikis has been an enormous success for collaborative editing on a large scale, as the online encyclopedia Wikipedia[2] has shown. In the myOntology project, we propose to use the infrastructure and culture of wikis for a collaborative and open ontology building environment.

[1] http://scholar.google.com, retrieved on February 22, 2007
[2] http://wikipedia.org/

1.2 Our Contribution

In this paper, we (1) argue that the use of social software will very much improve the state of the art in ontology engineering. We then sketch (2) design principles as well as (3) major components of the myOntology platform. As the project is in an early state, we present a (4) preliminary roadmap for the implementation of the platform. Finally, we (5) evaluate our approach by comparing it with traditional ontology engineering and (6) introduce the notion of horizontal and vertical ontology engineering.

1.3 Related work

Our work is related to the following streams of research:

Collaborative ontology engineering: Substantial literature on collaborative ontology engineering already exists. However, the approaches we know of do not put the Wiki editing approach in the center of building ontologies collaboratively. [4], [5] describe Tadzebao and WebOnto: Tadzebao is a system that supports asynchronous and synchronous discussions on ontologies. While we agree that allowing dialogue is important, we question that users are willing to spend the time to agree on a concept definition by a non-structured discussion. In our opinion, the support for achieving consensus must be more subtle. WebOnto, a Java based ontology editor, complements Tadzebao by supporting collaborative browsing, creation, and editing of ontologies. [6] describe the DILIGENT knowledge process where ontology evolution and collaborative concept mapping are applied to deal with conceptual dynamics of domains. The ontology editor Protégé[3] is also available in a Web version. OntoEdit [7] is a collaborative ontology editing environment that allows multiple users to develop ontologies in three phases: kick-off, refinement, and evaluation/maintenance. It ensures consistency by blocking the part of the ontology that is being edited. [8], [9], and [10] propose an approach to community-driven ontology matching that allows different individuals to create mappings.

Semantic Wikis: [11] allows annotating links, typing of pages, and context dependent content adaptation. Additionally, it displays related pages. [12] allows annotating links, typing of pages, and context dependent content adaptation. Additionally, it displays related pages. [13] have the objective to make the knowledge within Wikipedia, the online encyclopedia, machine-accessible by adding semantic information, e.g. by typing of links. They propose the use of semantic templates in order to simplify annotation for users. Platypus Wiki [14] describe only the similarities between collaborative ontology engineering and wikis. The approach described in this paper differs clearly from all these approaches, as most of them aim at augmenting existing wiki content with semantics. The goal of our approach is to *use* wiki technology to collaboratively build ontologies.

[3] http://protege.stanford.edu/

2 The myOntology approach

The myOntology approach towards ontology engineering clearly differs from traditional, engineering-oriented approaches. In this section, we describe our approach. In section 2.1, we define some design principles which reflect the myOntology philosophy. In section 2.2, we summarize the major components of the project. In section 2.3, we describe how existing technology contributes to the project.

2.1 Design principles

The goal of the myOntology project is to establish the theoretical foundations of collaborative, community-driven ontology building using wikis. The following design principles constitute the philosophy of myOntology:

Community grounding: The engineering-oriented ontology building approach, where a small number of ontology engineers constructs the representation of the domain of discourse and releases the results at some point in time to a wider community of users has several disadvantages: ontologies representing domains comprising a high degree of conceptual dynamics need to be changed often. A centralized approach will be too slow to appropriately reflect these changes, since missing entries cannot be added to the ontology by any user who reveals the need for a new concept, but instead have to be added by a small group of ontology engineers. This will at all times hinder ontology evolution and produce outdated thus not usable ontologies.

Furthermore, different individuals might have different views of a domain and therefore conflicts arise. The engineering-oriented approach forces users to commit to the view of a small group of ontologists. Our goal is not only to allow co-existence and interoperability of conflicting views but more importantly support the community in achieving consensus similar to Wikipedia, where one can observe that the process of consensus finding is supported by functionality allowing discussion.

Another disadvantage of the traditional, engineering-oriented ontology building approach is the lack of communication between ontology creator and user who cannot easily grasp the intention of a concept. As visualized in Figure 1, usually the user only has the serialization of the given ontology, which at best contains a textual description of the intention of the contained concepts in the form of non-functional properties.

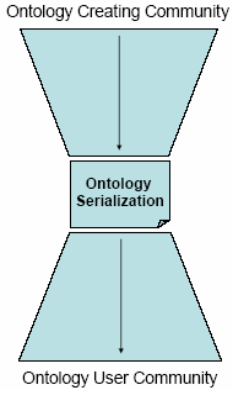


Fig. 1. The ontology perspicuity bottleneck [3]

A community based, de-centralized approach will reduce this problem, because ontology users can use the discussion that lead to the introduction of a concept as an additional hint to grasping the consensual meaning.

Ease of use: Existing traditional ontology engineering environments usually impose quite high entrance barriers on a user: a user with common Web-editing skills will not likely be able to create an ontology in e.g. Protégé[4] quickly. Social software offers the tools and paradigms in order to move from centralized towards de-centralized, community-grounded ontology building. Wikis allow many users to contribute easily with only basic Web-editing skills. However, the success of wikis also lies in many small but effective scripts that help the community build and maintain the corpus of knowledge, such as allowing discussion or the history function. We aim at developing small helper functionality that supports the community in developing the ontology.

Lightweight ontologies: The ontologies built with an open environment like myOntology might be rather simple models with a subsumption hierarchy. Though more expressive ontologies support more sophisticated reasoning we believe that also flat ontologies can be very useful. Even with a low degree of expressivity, such a framework would solve the problems described above with a focus on the first three bottlenecks as described in the previous section: conceptual dynamics, cost vs. benefit, and perspicuity [3]. Furthermore, one has to keep in mind that a high expressivity allowing for complex language constructs might overstrain and scare most users away. When defining a suitable meta-model for a wiki-based ontology building framework, a trade-off between expressivity and usability needs to be made.

[4] http://protege.stanford.edu

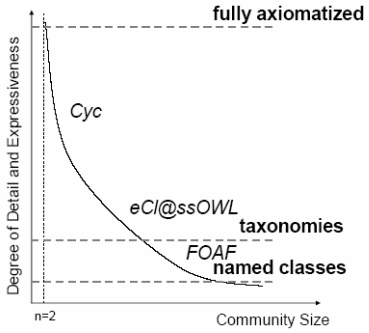


Fig. 2. The Expressivity-Community-Size Frontier [3]

[3] describes this as the expressivity-community-size frontier (Figure 2). It clearly shows that the more expressive an ontology is, the smaller is the user community as commitment costs are very high. Rather shallow and small ontologies such as FOAF have shown that ontologies have to comprise reasonable commitment costs.

2.2 Architecture

Addressing the problems delineated in the previous section involves divergent challenges, both within ontology engineering and beyond. In this section, we outline the major components of myOntology, which are visualized in Figure 3.

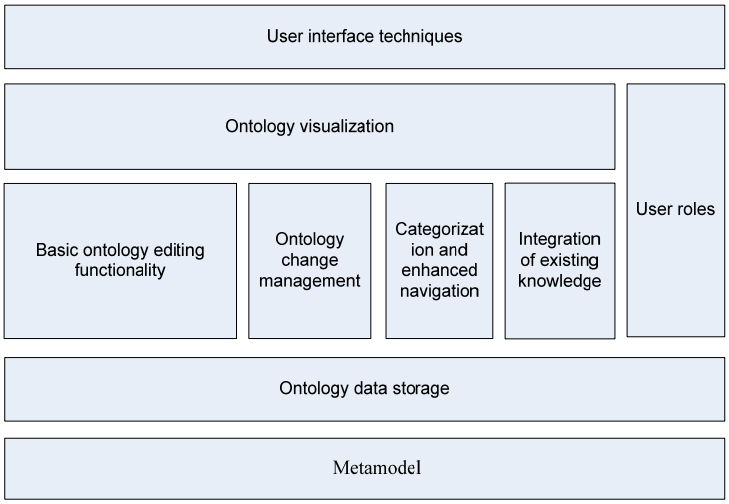


Fig. 3. myOntology components

Meta-model: We need to define an ontology meta-model that is suitable for a large audience. Obviously, non-expert users are not able to build highly axiomatized ontologies; as explained above flat ontologies can be useful as well. Additionally, reasoning support is desirable which comprises limitations concerning expressivity. The meta-model must support adding concepts, properties, and relations, as well as instances and several annotation properties. In order to support the upload of more expressive ontologies, elements that are not included in the meta-model will be preserved within annotation properties.

Ontology data storage: Ontology data as well as administrative data (e.g. user management) will be stored in a triple store using the myOntology ontology which represents the concepts and properties that are used within the environment. As myOntology will be open to the general public, the performance is especially important in order to preserve usability.

Basic ontology editing functionality: The focus especially in the first phase of the project lies on basic ontology editing functionality, such as adding and editing new classes and properties.

Ontology change management: Ontology change management comprises ontology evolution and versioning, as well as matching and mapping. In myOntology, we aim at community-supervised ontology change management: it is the community who track inconsistencies and remove them.

Categorization and enhanced navigation: In Wikipedia categories are used to enable better navigation and organization [15]. By very simple means, e.g. adding tags to definitions of ontology elements, a similar categorization system can be created in order to improve clarity as well as navigation additional to ontology browsing and concept search.

Integration of existing knowledge: In myOntology, we aim at integrating existing knowledge, such as references to Wikipedia articles. Furthermore, especially for e-commerce ontologies, the integration of eClassOWL [16] will allow much reuse of existing concepts. Dealing with homonyms and synonyms can be supported by using Wordnet [17]. Additionally, Google's mechanism for discovering spelling mistakes can add more value for the user.

Ontology visualization: In the collaborative ontology engineering paradigm it is extremely important that the meaning of a concept is obvious and easily understandable. In myOntology, ontology visualization techniques are emphasized additionally to a traditional, tabular view in order to help users understand the structure of an ontology, such as tag clouds and topic maps.

User roles: In myOntology, multiple kinds of roles which are necessary achieve consensus while editing and modifying an ontology are specified. We distinguish between four types of users: first, content consumers simply browse or use an ontology. Second, content providers regularly add new content. Third, content reviewers play an active role as well by reviewing existing content and participating in discussions. Fourth, super users are a few selected moderators, who act as administrators to the whole process and can, as a last resort, overrule the rest of the community. Mechanisms to assign user roles to users could be, e.g., an ontology modeling test, where users have to prove their abilities in ontology building.

User interface techniques: The importance of the design of the user interface is obvious as the audience of the project is very broad and non-technical. Most academic

prototype implementations neglect the design their user interface. We will aim at building a nice and easy-to-use user interface based on existing work on interfaces, such as [18]. Additionally, we propose the use of multimedia elements. A natural language description of a concept supported by a picture conveys much more meaning than only text and improves disambiguity of informal concept definitions.

2.3 Contribution of existing technology components

MyOntology is an interdisciplinary project involving many research areas. We will make use of existing technology and state-of-the-art work in the most areas. For the myOntology meta-model a subset of OWL DLP [19] will be extended with some constructs from SKOS [20]. In order to support round tripping, more complex ontology elements that are not included in the meta-model are preserved by storing them using annotation properties. For storing ontology data, many different triple stores already exist. [21] present a detailed comparison of existing approaches. Substantial work has already been done in the area of traditional ontology development environments, such as Protégé. These tools provide excellent environments for skilled users allowing the creation of ontologies with a varying degree of expressivity. Existing ontology building tools will serve as a model for myOntology when it comes to basic ontology editing tasks, such as adding new classes, editing existing elements, etc. Handling ontology changes is probably the most complex challenge for myOntology. We will combine existing approaches with a community-supervised style of change management: similar to Wikipedia the community tracks inconsistencies and aligns concepts. Furthermore, we will exploit existing visualization techniques: implicit information contained in an ontology, such as the underlying structure of a data model or which instances are most closely connected is all contained in a graph. This information, though, is difficult, if not impossible, to extract from a text-based reading of the data. MyOntology will use techniques such as tag clouds and semantic networks.

3 Implementation

The myOntology project is in its early implementation phase. We follow the rapid prototyping paradigm due to the following reasons: (1) first results are visible immediately and work done can be verified instantly and (2) industrial partners can constantly check the development through an early quality assurance. First design decisions have been made: As a programming paradigm Java JSP will be used. PHP (like MediaWiki) was considered, however, Java gives more freedom when it comes to extensibility and the implementation of small helper scripts. Furthermore, Sesame is used as a triple store in order to store both, ontology data as well as administrative data. The first version of the prototype will be a wiki-based platform for browsing an ontology based on a minimal ontology meta-model. This meta-model supports classes (i.e., concepts / categories), attributes (i.e., slots for data type or object values assigned to classes), value categories as a special type of ontology classes, value instances of those value categories, and the “subclass Of” relation. The first prototype

will allow users to suggest extensions or changes to any ontology element. Such change requests are pre-classified according to the context and semi-automatically processed by a privileged domain expert. In detail, users can recommend adding new classes, attributes, value types, as well as open feedback.

4 Evaluation

As the myOntology project is in an early stage and implementation work has only just begun, we evaluate our approach by comparing traditional ontology engineering to the myOntology approach. Additionally, we show the difference between *horizontal* and *vertical* ontology maintenance and why myOntology focuses on the support of horizontal ontology maintenance.

4.1 Traditional ontology engineering vs. the myOntology approach

The criteria for the evaluation are: (1) As shown by [22], many domains, especially in e-commerce, comprise a high degree of conceptual dynamics. *Timeliness* describes whether an ontology is up-to-date and hence useful. (2) *User participation* is an indicator how many individuals can contribute to the ontology evolution and especially if the control over the evolution is with the actual user community. (3) As ontologies are community contracts, the degree of *community grounding* depends on how agreed upon the representation of a domain is. (4) The *expressivity* of an ontology can range from flat collections of terms to abundantly axiomatized ontologies. (5) *Consistency:* ontology inconsistencies occur when the ontology is changed. Both design approaches comprise different risks for inconsistency.

	Traditional ontology engineering	**myOntology approach**
Timeliness	For an individual engineer or a group of engineers it is (a) more expensive and (b) more complex to keep the ontology up to date. Hence, ontologies maintained in a traditional ontology engineering approach, are more likely to be outdated.	A big community can keep up with the pace of conceptual dynamics more easily. This phenomenon can also be observed in Wikipedia. Therefore, myOntology will produce more up-to-date ontologies. This is a crucial feature in business domains.
User participation	In the engineering-oriented approach, only a small group of ontology engineers is involved. Users can only contribute by suggesting changes e.g. per e-mail or fax. This does not hinder	In the myOntology approach the actual users of an ontology can contribute to and control the evolution of an ontology. This makes a commitment much easier for them.

	ontology evolution but also consumes a lot of resources.	
Community grounding	Ontologies created in the traditional manner represent the view of the few ontologists working on the specification. Therefore, misconceptions are likely as well as a cleavage between the ontology and what the view of the community is.	Ontologies created with myOntology are real community contracts: like in Wikipedia, the community agrees on a specification supported by different functionality, such as discussion and history.
Expressivity	Depending on the skills of the engineers, in the traditional approach highly axiomatized, expressive ontologies can be created.	MyOntology will produce rather lightweight ontologies as most users can not be expected to be able to add axioms. However, this is not only a disadvantage: as shown by [22], a simpler ontology will have a bigger user community (which is desirable).
Consistency	In traditional ontology building, the resulting ontologies are more likely to be consistent as only a small group of skilled ontologist will work on the specification.	The more users the more likely inconsistencies occur. On the other hand, these inconsistencies can be tracked by the users themselves, like in Wikipedia.

Table 1. Traditional ontology engineering vs. the myOntology approach

4.2 Horizontal vs. vertical ontology maintenance

Regarding the expressivity of ontologies produced with myOntology, they will be rather lightweight. Too much expressivity will overstrain users and therefore hamper the creation of ontologies. In the following section, we describe the relation between horizontal and vertical ontology maintenance and the expressivity of ontologies and why myOntology can be described as a horizontal approach.

We distinguish between vertical and horizontal ontology maintenance (Figure 4): horizontal ontology maintenance can be understood as extending an ontology by concepts and properties but not in the level of detail or axiomatization. Vertical ontology maintenance emphasizes extending an ontology by axioms.

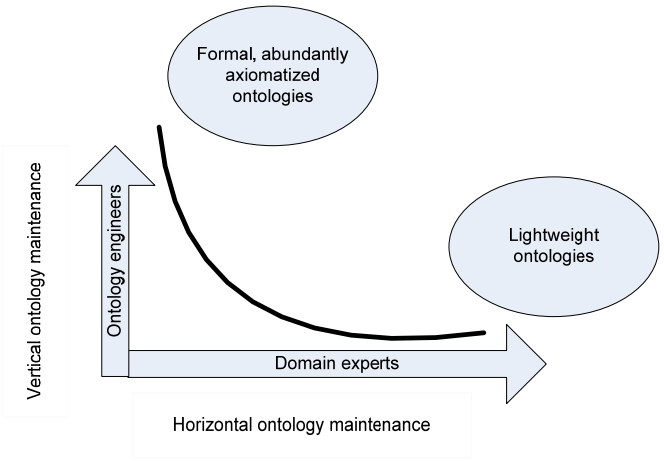


Fig. 4. Horizontal and vertical ontology maintenance

While in horizontal ontology maintenance, ontologies are rather shallow and lightweight, vertical ontology maintenance produces formal ontologies. Users will be able to create ontologies with a clear subsumption hierarchy with myOntology. In an expert mode it will be possible to add more complex constructs. However, the majority of ontologies created will be rather lightweight. The target groups for the project are the research community as well as domain experts with only basic Web editing skills, which makes myOntology a horizontal maintenance tool.

5 Conclusion

Ontologies are widely regarded as the backbone of the Semantic Web. However, only few ontologies can be found. Some reasons for this were outlined in the first section. The myOntology project described in this paper is supposed to enable more users to participate in creating and maintaining ontologies. Though these ontologies might not be highly axiomatized, they will be very useful to describe domains that can benefit from deploying ontologies, such as e-commerce. We introduce the notion of horizontal ontology maintenance opposed to vertical ontology maintenance, where myOntology is rather a horizontal approach. Open ontology engineering must have proper technical foundations but social and usability aspects must be considered as well. Wikis are social software that recently has been proven efficient and popular. Providing users with a usable tool that supports the community to establish community contracts on ontology definitions will result in more simple but useful ontologies that will be actually used in Web applications. The myOntology project aims at exploiting the collective intelligence of a community for ontology engineering.

Acknowledgments. This work has been funded by the FFG project myOntology (contract number 812515).

References

1. Fensel, D., *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*. 2nd ed. 2004, Berlin etc.: Springer.
2. Ding, Y., et al., *The Semantic Web - Yet Another Hip?* Data & Knowledge Engineering, 2002. **41**(2-3 (June 2002)): p. 205-227.
3. Hepp, M., *Possible Ontologies: How Reality Constrains the Development of Relevant Ontologies.* IEEE Internet Computing, 2007. **11**(7): p. 96-102.
4. Domingue, J. *Tadzebao and WebOnto: Discussing, Browsing, and Editing Ontologies on the Web*. in *11th Knowledge Acquisition for Knowledge-Based Systems Workshop*. 1998. Banff, Canada.
5. Aitchison, J., A. Gilchrist, and D. Bawden, *Thesaurus construction and use: a practical manual*. 4. ed. 2000, London: Aslib IMI.
6. Vrandecic, D., et al., *The DILIGENT knowledge process.* Journal of Knowledge Management, 2005. **9**(5): p. 85-96.
7. Sure, Y., et al. *OntoEdit: Collaborative Ontology Engineering for the Semantic Web*. in *International Semantic Web Conference 2002 (ISWC 2002)*. 2002. Sardinia, Italia.
8. Gabel, T., Y. Sure, and J. Voelker, *KAON - Ontology Management Infrastructure* in *SEKT Deliverable D3.1.1.a*. 2004.
9. Zhdanova, A. and P. Shvaiko. *User-driven Ontology Matching.* in *3rd European Conference (ESWC2006)*. 2002. Madrid, Spain.
10. Zhdanova, A. and P. Shvaiko. *Community-driven Ontology Matching*. in *Wiki meets Semantics workshop at the ESWC2006*. 2006. Budva, Montenegro.
11. Dello, C., E. Paslaru Bontas Simperl, and R. Tolksdorf. *Creating and using semantic content with Makna.* in *Wiki meets Semantics workshop at the ESWC2006*. 2006. Budva, Montenegro.
12. Schaffert, S. *Semantic Wikipedia.* in *1st international workshop on World Wide Web (WWW2006)*. 2006. Manchester, UK.
13. Völkel, M., et al. *Semantic Wikipedia*. in *15th international conference on World Wide Web (WWW2006)*. 2006. Edinburgh, Scotland.
14. Bao, J. and V. Honavar. *Platypus Wiki: a Semantic Wiki Wiki Web*. in *3rd International Workshop on Evaluation of Ontology-based Tools (EON2004)*. 2004. Ancona, Italy.
15. Wikipedia. *Wikipedia Categorization*. 2007 [cited 2007 March 26]; Available from: http://en.wikipedia.org/w/index.php?title=Wikipedia:Categorization&oldid=117808880.
16. Hepp, M. *eCl@ssOWL. The Products and Services Ontology*. 2006 [cited 2006 May 20]; Available from: http://www.heppnetz.de/eclassowl/.
17. Fellbaum, C., *Wordnet: an electronic lexical database*. 1998: The MIT Press.
18. Tidwell, J., *Designing Interfaces.* 2nd ed, ed. E. Odewahn and M. O'Brien. 2006: O'Reilly. 331.
19. Vrandecic, D., et al., *DLP - An introduction*, in *Technical report*. 2005, AIFB, Universität Karlsruhe: Karlsruhe.
20. W3C. *Simple Knowledge Organisation System (SKOS).* 2005 [cited 2005 Nov 30]; Available from: http://www.w3.org/2004/02/skos/.
21. Berners-Lee, T. *URI References: Fragment Identifiers on URIs.* 1997 [cited 2004 08.11.2004]; Available from: http://www.w3.org/DesignIssues/Fragment.html.
22. Adida, B. and M. Birbeck. *RDFa Primer 1.0. Embedding RDF in XHTML. W3C Working Draft 16 May 2006.* 2006 [cited 2006 November 27]; Available from: http://www.w3.org/TR/xhtml-rdfa-primer/.

Author Index