A Web 3.0 Approach for Improving Tagging Systems

Alexander Kreiser
IBM Research and Development
Schönaicher Straße 220
71032 Böblingen, Germany
akreiser@googlemail.com

Andreas Nauerz
IBM Research and Development
Schönaicher Straße 220
71032 Böblingen, Germany
andreas.nauerz@de.ibm.com

Fedor Bakalov
Friedrich-Schiller University of Jena
Ernst-Abbe-Platz 1-4
07743 Jena, Germany
fedor.bakalov@uni-jena.de

Birgitta König-Ries
Friedrich-Schiller University of Jena
Ernst-Abbe-Platz 1-4
07743 Jena, Germany
birgitta.koenig-ries@uni-jena.de

Martin Welsch
IBM Research and Development
Schönaicher Straße 220
71032 Böblingen, Germany
martin.welsch@de.ibm.com

ABSTRACT

In Web 2.0 systems tagging has become one of the most popular techniques to allow users (and entire user communities) to categorize content autonomously. But, current tagging systems have their flipsides, though: synonyms and polysems lead to littered tag spaces making it difficult for users to find relevant content. Users suffer from retrieving content actually not being of interest or, vice versa, from not retrieving content that actually would be of interest when exploring the tag space. Moreover, in current tagging systems no relations between tags are modeled. Thus, recommending related tags (or content) is not possible.

In this paper we present an approach allowing users, i.e. the community, to collaboratively model relations between tags. We provide UI components allowing to model these relations which are then stored in a SKOS-based ontology which can be leveraged for content recommendations. Giving the community the power to consolidate tags and to relate tags to each other and, at the same time, storing these relations in ontologies is our Web 3.0 approach to solve tag space littering problems and to issue tag-based recommendations.

The concepts presented are being prototypically implemented within IBM's WebSphere Portal and can be presented in a live demo at the workshop.

Categories and Subject Descriptors

I.2.4 [**Knowledge Representation Formalisms and Methods**]: Predicate logic,Relation systems,Semantic networks; H.3.3 [**Information Search and Retrieval**]:

Keywords

Web 2.0, Web 3.0, Semantic Web, Tagging, Recommender

1. INTRODUCTION

The recent popularity of collaboration techniques on the Internet, particularly tagging and rating, provides new means for both semantically describing web content as well as for reasoning about users' interests, preferences and contexts. It can add valuable meta information and even lightweight semantics to web resources. Tagging allows non-expert users to develop folksonomies that categorize content available in the system.

Unfortunately most current tagging systems have two main drawbacks:
First, synonyms and polysems cannot be easily detected automatically and thus litter the tag space. Synonyms lead to multiple tags that all can have the same meaning, either because they are only a morphological variation (*apple* vs. *apples*) or semantically similar (*baby* vs. *infant*). Polysems lead to single tags that can have multiple meanings (*apple* can refer to the fruit or to the company Apple). From a user perspective the two problems might manifest themselves as follows: A user Alice and a user Bob might both apply the tag *apple* to some resources, but Alice might refer to the fruit whereas Bob might refer to the computer manufacturer. When Bob is later doing information retrieval by selecting the tag *apple* he receives a lot of "irrelevant noise" as he is also presented resources that have to do with the computer manufacturer. This problem is referred to as *low precision* problem. In the second scenario both users might want to tag resources providing information about the United States. Alice might tag these resources with *USA*, Bob with *United States*. When doing information retrieval Alice, might miss the resources tagged by Bob and Bob might miss the resources tagged by Alice just because the semantic relatedness of these tags remains invisible for the system. This problem is referred to as *low recall* problem. Current approaches to solve these problems include the application of normalization and stemming algorithms (cp. e.g. [8]) to prevent littering due to synonyms and the application of multiple tags to single resources to prevent littering due to polysems. But both approaches have their limits.
Second, tags are flat lists of words of uncontrolled vocabularies not having any relations. Thus, current tagging systems can hardly recommend users with related content. Being able to recommend users with more generally available con-

cepts (e.g. a user interested in making *Spaghetti* might be interested in making *Pasta* in general, too), with more specific concepts (i.e. recommending information about *Spaghetti*, *Farfalle*, etc. when a user is searching for information about *Pasta*) or just with related concepts (i.e. recommending information about *cat food* when a users reads material about *cats* in general) are highly appreciatable features in such large system we deal with.

In this paper we present a Web 3.0 approach for solving the problems just mentioned. First, we allow users, i.e. the community, to augment tags to make them less ambiguous. Second, we allow users to collaboratively model relations between tags which can then be leveraged for content recommendations. We provide the community with UI components to model these relations which are then stored in a SKOS-based ontology. Giving the community the power to consolidate tags and to relate tags to each other and, at the same time, storing these relations in ontologies is our Web 3.0 approach to solve tag space littering problems and to issue tag-based recommendations.

2. RELATED WORK

Several approaches aim to improve tagging systems by using semantically rich tags instead of flat keywords. We will refer to these systems as semantic tagging systems. These approaches can be split into two groups based on the strategies they use:

1. Semantifying already existing tags of a tagging system
2. Enabling a community to annotate resources with semantically rich tags instead of flat keywords

With the first strategy flat keywords of existing tagging systems (like del.icio.us) are enriched with semantics. This is often an automatic process in which tags are mapped to concepts from an ontology or relations between tags are derived from the folksonomy structure. The system *TagOnto* proposed in [10] automatically maps tags from a social tagging system to entities in domain ontologies. In [1] tags are also mapped to ontology concepts in order to improve the retrieval process and recommend related content. Heymann and Garcia-Molina introduced an algorithm in [7] to transform a set of flat tags into a hierarchical taxonomy. Angeletou et al. ([11]) create semantic relations between tags by leveraging the semantics stored in public ontologies.

There were also efforts of conceptualizing and implementing a new kind of tagging systems, where users define the meaning of a tag when it is applied. Hence, tags are no longer keywords but references to entities in semantic repositories. In many of these systems Semantic Web technology was used to store the meaning of a tag and semantic relations between tags. The following two paragraphs describe two types of semantic tagging systems and their related work. With the first type existing public ontologies were used while with the second one the tagging community creates the tag ontology itself.

SemKey - a semantic collaborative tagging system described in [5] - utilizes Wikipedia and WordNet[1] to disambiguate keywords used for tagging while the actual tags then refer to Wikipedia pages. The social bookmarking website *Faviki* enables the annotation of bookmarks with DBpedia[2] concepts. The *entity describer*[3] is an add-on to the tagging system of Connotea[4] where tags refer to entities from the freebase[5] ontology. In [2] Passant and Laublet propose *MOAT*, a client-server framework where the server provides services for term disambiguation and finding matching concepts in several public knowledge bases (e.g.: DBpedia, Yahoo! Geonames).

With the second type, the tag ontology is created by the same community that uses it for tagging. That way, the semantic data is tailored to a tagging community and kept as small as necessary. In [3] Braun et al. describe two prototypes of semantic tagging systems where the creation of semantic data is merged into the tagging process: *SOBOLEO* and the *IMAGINATION* project. *RichTags* ([4]) is the name of a social bookmarking system resulting from the master thesis of Fountopoulos where users build and extend a SKOS ontology collaboratively. In *Fuzzzy* ([9]) a community creates its own ontology using Topic Maps[6] and is then able to bookmark sites with the created topics.

3. OUR APPROACH

In this section we describe our approach for solving the problems of tagging systems by combining Semantic Web technology with Web 2.0 interface components. Section 3.1 describes our ontology design and how users can provide semantic relations between tags. In Section 3.2 we give a brief overview over the web interface and its components. Section 3.3 shows the current system architecture of our prototype.

3.1 Linked Knowledge Islands

We rely on Semantic Web technology regarding the storage of tags and relations. SKOS (Simple Knowledge Organization System, [6]) is a W3C candidate recommendation for modeling thesauri and loose taxonomies in RDF format and builts upon OWL[7]. Like [3] and [4] we chose the SKOS vocabulary to store our tag ontology. In our system tags are instances of `skos:Concept`, have multiple labels and a definition that helps to disambiguate tags with identical labels. Users are able to apply a small set of semantic relations: `skos:broader` and `skos:narrower` to model a loose taxonomy and `skos:related` to create associative links between tags. Broader and narrower relations can help to solve the abstraction level problem: A search for resources annotated with *Animal* should also return resources annotated with *Cat*, *Dog*, *Bird*, etc. The related property can be leveraged for tag and content recommendation. This is useful to extend search parameters or reformulating a search query but also to suggest additional tags during the tagging process.

We chose SKOS to model the tag ontology since in our opinion the small predefined set of semantic relations is suitable

[1] http://wordnet.princeton.edu/

[2] http://dbpedia.org

[3] http://www.entitydescriber.org/

[4] Online reference management for clinicians and scientists, http://www.connotea.org/

[5] http://www.freebase.com/

[6] http://topicmaps.org/

[7] Web Ontology Language, http://www.w3.org/2004/OWL/

for a big community. Users are able to create loose taxonomies and general associative links between tags without having to be experts in ontology engineering. Also, our ontology is compatible with other SKOS-based knowledge systems which enables us to use existing SKOS ontologies as a base or to cover certain knowledge domains.

Roy Lachica states in [9] that "users already have a mental representation of the world and have no need to externalize this view by entering their world view into the system." This conclusion is drawn from the lack of contribution he experienced testing the *Fuzzzy* social bookmarking system where users collaboratively build an ontology. Another observation made by Lachica was that in "...several cases where users do not agree with tag-relations that have been created by others but they take no action to correct it by voting or other means"([9]). Having one collective knowledge model is a difficult task to achieve since the individual mental models of individual users will always diverge to some degree. Disagreement on labels and relations used within the model can be a major issue lowering user experience and motivation to participate. In our linked knowledge islands approach users are able to model their private tag space and draw immediate benefits from the semantic relations, e.g. by exploiting the taxonomy and associations to automate steps in their retrieval process.

This is achieved by giving each user the opportunity to structure and label his tags the way he wants. Additionally, a user can map any of his tags to tags from other private models if both tags refer to the same concept from the real world. This mapping is stored as `skos:exactMatch` property in the ontology. Everyone in the community can have the benefits of the collective knowledge model if he wants, but is also allowed to stay within his own model. Every user has a concept scheme (`skos:ConceptScheme`) which contains all tags created by him. Concept schemes in SKOS were designed to aggregate concepts when dealing with data from different knowledge organization systems - in our case the different mental models of the users. Since the `skos:exactMatch` property is symmetric and transitive the separate knowledge islands are quickly connected to a big network. If e.g. Bob links his tag *Semantic Web* in his model to Alice's tag *Web 3.0* in her model he is able to exploit parts of her knowledge model as well. If Alice has previously linked her tag *Web 3.0* to a tag in Carl's model, Bob can exploit Carl's model as well and vice versa (Figure 1).

3.2 Web 2.0 Graphical User Interface

The simplicity and flexibility of folksonomies is one of the main reasons for the success of social tagging systems today. Specifying the semantics of a tag can solve some of the problems of tagging systems and therefore improve user experience. But it also means having to extend the tagging process which results in a work overhead for the user. In order to keep this overhead small, we tried to design a user interface that allows the community to perform the necessary tasks fast and easily. One aspect of Web 2.0 is the improved usability provided by Rich Internet Applications (RIA) using AJAX technology and a more desktop-like look and feel. Our web interface consists of several widgets built

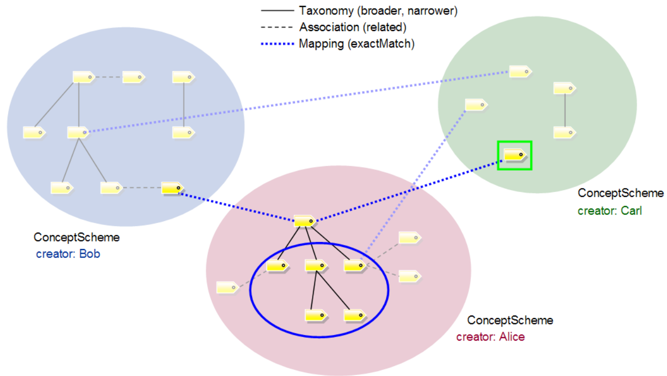

Figure 1: Narrower tags are retrieved from a mapped tag

with the Dojo Toolkit[8] which interact with the web services of our system.

Locating and Disambiguating Tags

Tags can be found by typing a term into a type-ahead enabled combo box which in turn displays a list of tags with matching labels. The user can then navigate through that list and obtain additional information like the tag definition to disambiguate tags with similar or identical labels. For an even faster disambiguation on first sight one of the broader tags is displayed in brackets (Figure 2). This process helps the user to locate a suitable tag for her intends and lets her specify the exact meaning of the tag she applies to a resource. If no suitable tag can be found the user is free to create a tag in her private model by providing a preferred label and a definition. Optionally, other tags that are broader or narrower than or related to the new tag can be declared, integrating the new tag directly into the taxonomy.

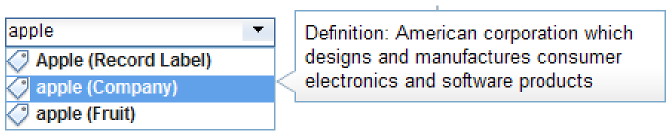

Figure 2: Type-ahead combo box to lookup tags

Providing Semantic Relations

Within the user interface the semantic tags always have the same visual representation and can be dragged and dropped to perform certain actions. If one tag is dropped onto another tag a context menu appears and the user can add a semantic relation between these two tags. For instance, if a user finds a tag from another user that refers to exactly the same concept as her tag but with a slightly different label, she can map these two tags using the context menu (see Figure 3). After annotating a resource, the semantic tagging application confirms the tagging and displays semantic tags other users have previously applied to this resource. If the user notices that a tag another user has applied is related to or about the same thing as one of her tags she can quickly provide this semantic relation using drag and drop and the context menu. This was one of our main objectives: Modeling semantics should be embedded in the tagging process.

[8]http://www.dojotoolkit.org/

Furthermore, tags can be dropped into tag bags which are visual representations of containers that hold a set of tags (e.g. a user can put his favorite tags into one single tag bag).

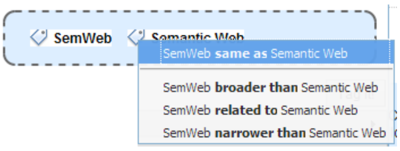

Figure 3: Adding semantic relations via drag and drop

Browsing the Taxonomy and Related Tags
In several scenarios it would be valuable to browse the taxonomy and relations defined in the tag ontology. For instance, when selecting semantic tags from the ontology in order to tag a resource, the most specific tag should be used instead of the most general. By exploring the tag model taxonomy and drilling down into narrower tags the user is able to select the most specific semantic tags for a resource. With the help of our browsing widget users can explore a visualization of the tag model. In the current prototype, the hierarchical relations of a tag can be explored with a tree view (Figure 4). Narrower tags are shown as child nodes in the tree while broader tags are visualized as parent nodes. Unfortunately, the tree visualization has obvious drawbacks; multiple broader tags can not directly be visualized and related tags aren't displayed at all. In our current implementation this information can be looked up by right-clicking on a tree node where related and broader tags are listed in a context menu.

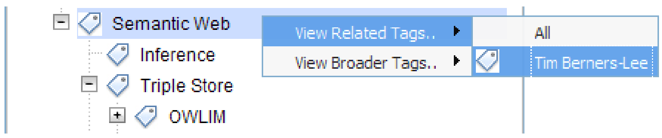

Figure 4: Browsing widget with the tree view

Therefore, we are experimenting with different methods of tag model visualization. A conceivable replacement for the tree view could be a simple graph visualization (Figure 5). Tags are displayed as nodes and straight lines between the nodes depict semantic relations where each semantic relation has its own color. By clicking on a node all semantic relations to directly related tags could be dynamically loaded and visualized. In that way a user can explore the tag space visually until she finds the information she needs. Since this is a rather uncommon type of interface user experience and acceptance are still to be evaluated.

Finally, leveraging the knowledge about relations between tags (and, implicitly, about resources) we can also automatically recommend users with related content as outlined before.

3.3 Architecture

The current system prototype is being implemented on IBM WebSphere Portal technology with a separate component

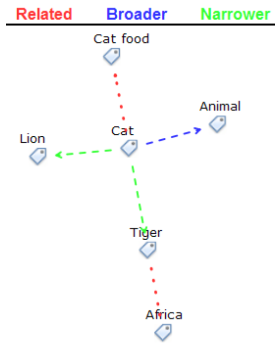

Figure 5: Graph visualization of the tag model

containing the knowledge base. The whole system architecture is loosely coupled using web services to access the individual components. Our choice regarding the semantic repository fell on openRDF sesame[9] since we considered it to be one of the most mature non-commercial RDF repositories. To enable OWL inferencing (which is necessary for the SKOS properties we use), swiftOWLIM[10] was integrated with sesame as Storage And Inference Layer (SAIL). A web service layer on top of the RDF repository adds convenient and simplified access to the semantic data which is used by the user interface widgets. Additional, customized SPARQL queries can be run against the repository making it possible for other applications to use the stored semantics.

Tags and their semantic relations are kept in the repository while the tagging system just stores URI references to the tags. The loose coupling makes it possible to swap the underlying tagging system or tagging API which takes care of annotating resources and retrieving resources by tag reference.

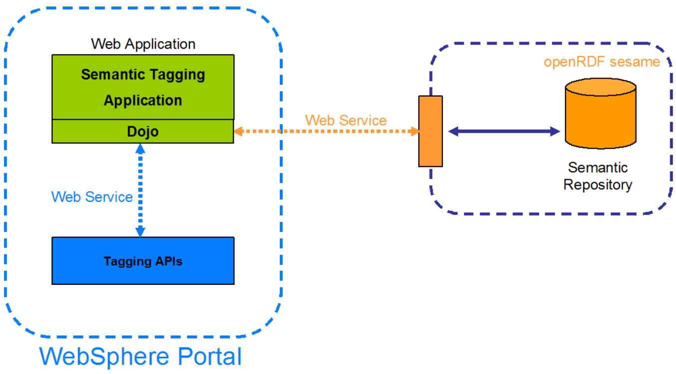

Figure 6: Abstract System Architecture

4. CONCLUSION AND FUTURE WORK

In this paper we have described a Web 3.0 approach to collaboratively modeling semantic relations among tags. Our approach solves a number of problems associated with the traditional tagging approach that leverages flat lists of unrelated tags. First, users are enabled to create tag hierarchies

[9] http://openrdf.org
[10] http://ontotext.com/owlim/

that fit their mental models, hence better organize content of their interest. Second, the semantic relations among tags can be used for improving the information retrieval process and recommending related content. Finally, our approach solves the tag disambiguation and tag space littering problems.

The ideas presented in this paper are being prototypically implemented in IBM's WebSphere Portal. Upon completion of implementation, we plan to evaluate usability of the proposed tagging process as well as the value of improvements that the collaboratively created tag ontologies provide for content recommendation. Also in the future we plan to extend our approach to let users collaboratively create more semantically rich OWL ontologies.

5. REFERENCES

[1] Alexandre Passant. Using ontologies to strengthen folksonomies and enrich information retrieval in weblogs. In *Proceedings of the First International Conference on Weblogs and Social Media (ICWSM)*, Boulder, Colorado, 2007.

[2] Alexandre Passant and Philippe Laublet. Meaning of a tag: A collaborative approach to bridge the gap between tagging and linked data. In *Proceedings of the WWW 2008 Workshop Linked Data on the Web (LDOW2008), Beijing, China*, 2008.

[3] S. Braun, A. Schmidt, and A. Walter. Ontology maturing: A collaborative web 2.0 approach to ontology engineering, 08.05.2007.

[4] G. Fountopoulos. *RichTags: A Social Semantic Tagging System: Dissertation for MSc Web Technology*. PhD thesis, University of Southampton, Southampton, 20 December 2007.

[5] A. Marchetti, M. Tesconi, F. Ronzano, M. Rosella, and S. Minutoli. Semkey: a semantic collaborative tagging system.

[6] A. Miles and S. Bechhofer. Skos simple knowledge organization system reference, 29.08.2008.

[7] P. Heymann and H. Garcia-Molina. Collaborative creation of communal hierarchical taxonomies in social tagging systems, 2006.

[8] M. F. Porter. An algorithm for suffix stripping. pages 313–316, 1997.

[9] R. Lachica and D. Karabeg. Metadata creation in socio-semantic tagging systems: Towards holistic knowledge creation and interchange. In L.M. Garshol and L. Maicher, editors, *Scaling Topic Maps. Topic Maps Research and Applications 2007*, Lecture Notes in Computer Science. Springer, 2007.

[10] Silvia Bindelli, Claudio Criscione, Carlo Curino, Mauro L. Drago, Davide Eynard, and Giorgio Orsi. Improving search and navigation by combining ontologies and social tags. In Robert Meersman, Zahir Tari, and Pilar Herrero, editors, *OTM Workshops*, volume 5333 of *Lecture Notes in Computer Science*, pages 76–85. Springer, 2008.

[11] Sofia Angeletou, Marta Sabou, Lucia Specia, and Enrico Motta. Bridging the gap between folksonomies and the semantic web: An experience report. In *Bridging the Gep between Semantic Web and Web 2.0 (SemNet 2007)*, pages 30–43, 2007.