

Reducing Ambiguity in Tagging Systems with Folksonomy Search Expansion

Jeff Z. Pan, Stuart Taylor, and Edward Thomas

Dept. of Computing Science
Univ. of Aberdeen
Aberdeen, UK

Abstract. Search facilities are vital for folksonomy (or social tagging mechanism) based systems. Although these systems allow great malleability and adaptability, they also suffer from problems, such as ambiguity in the meaning of tags, flat organisation of tags and some degree of unstabilising factor on consensus about which tags best describe some certain Web resources. It has been argued that folksonomy structure can be enhanced by ontologies; however, as suggested by Hotho et al., a key question remains open: how to exploit the benefits of ontologies without bothering untrained users with its rigidity. In this paper, we propose an approach to address the problem of ambiguity in tagging systems by expanding folksonomy search with ontologies, which are completely transparent to users. Preliminary implementations and evaluations on the efficiency and the usefulness of such expansions are very promising.

1 Introduction

Web 2.0 applications are emerging providers of a new type of Web resources. On the one hand, more and more Web resources become available in popular folksonomy (or social tagging mechanism) based systems, such as Flickr¹, YouTube² and del.icio.us³. For example, in April 2008 alone, there are about 2.5 million photos geotagged⁴ in Flickr. On the other hand, mashup applications combine Web resources from more than one folksonomy based system into various integrated experience [11]. These two observations indicate that search facilities are vital *both* within folksonomy based systems *and* across folksonomy based systems.

There are advantages and disadvantages of the tagging approach [9]. The folksonomy based systems allow great malleability and adaptability, and the simplicity of adding and using tags allows them to be more accessible to users than well defined (more complex) classification systems. However, these systems also suffer from problems [17], such as ambiguity in the meaning of tags and flat organisation of tags (see Section 3.1).

For example, in the results from a search on YouTube for “Focus” (the Dutch progressive rock band), only 3 of the top 20 items in the list are relevant to that band. The

¹ <http://www.flickr.com/>

² <http://www.youtube.com/>

³ <http://del.icio.us/>

⁴ Geotagging is the art of adding location information to things like a photo, according to Flickr.

remaining 17 search results relate to alternative meanings of the word “focus”. This simple example illustrates that ambiguity in the meaning of tags is a significant problem for domain specific folksonomy searches. Furthermore, for each of the 3 relevant results, the tags list for each result contained few other relevant tags than “Focus” and the name of the song in the video. In every case, information on at least of: the type of music, the album, the band members and other relevant information was omitted from the tag list.

Halpin et al. expressed a major concern of the folksonomy approach, i.e., whether or not in the folksonomy based system consensus about which tags best describing some certain Web resources can be stabilised, so that such best tags are used most often with the folksonomy based systems [9]. There could be a serious problem if no coherent categorisation scheme can emerge at all from collaborative tagging. This is a crucial point because ranking techniques employed by major Web search engines today are, to a large extent, influenced by people’s opinion of Web resources [10]. Although Halpin et al. concluded that tagging distributions *tend to* stabilise into power law distributions, the process of stabilisation (w.r.t. certain contexts) depends on the number of active users and to some extent on the temporal duration of the tagging process, according to their empirical study. Concrete predictions are not (yet) available on the number of active users and duration of time needed to complete the stabilisation process. The situation of searching across multiple folksonomy based system could be even worse, since different styles and environments provided by different systems could make it harder to reach stabilised consensus. Some might even argue that it is impossible to reach such Web scale stabilised consensus, as consensus from different systems might be conflicting with each other.

There has been significant interests on combining Semantic Web and Web 2.0 [3,6,12]. It has been suggested [10,17,1,4] that Semantic Web technologies could help improve the internal structure of folksonomies. Passant proposed to encourage users to link tags with concepts and individuals in ontologies [17]. However, as suggested by Hotho et al. [10], a key question remains open: how to exploit the benefits of ontology without bothering untrained users with its rigidity.

In this paper, we propose a novel approach to using ontology entailments to expand folksonomy search, without bothering untrained users to use ontologies directly. More precisely, the paper makes the following major contributions:

1. Our approach allows Web application developers to exploit the benefits of ontology without bothering untrained users with its rigidity. *Ontology transparency* is achieved by (i) automatic associating keywords with classes, properties and individuals in domain ontologies (Section 4.1) and (ii) relating the search keyword(s) with keywords based on the structure of domain ontologies (Section 4.2).
2. We propose a general framework of ontology-based folksonomy search expansion methods for reducing ambiguity (Section 4), based on some use cases and observations on search expansions (Section 3). For example, a search for “focus” could be expanded to include (some) album titles, songs titles, live performance venues, as well as keywords related to the domain in general. Hence, more meaningful results are returned to users than making use of the tagging systems directly. An interesting feature of the proposed methods is that all the required ontology entailments

are included in OWL2 QL semantic approximations [15] of given domain ontologies (Section 2). This suggests that theoretically the proposed methods can be very efficient.

3. We report the implementations in our ontology infrastructure of the proposed automatic keyword-ontology association and the proposed ontology-based folksonomy search expansion methods (Section 5). Preliminary evaluations indicate that these expansion methods are efficient even for some very large ontologies (Section 6).
4. We present a semantic mashup application, called MusicMash2, to demonstrate how the above ontology infrastructure could enhance current Web 2.0 application and experience (Section 5.2). MusicMash2 provides music domain-specific search features to retrieve data from folksonomy based systems, offering search by genre, artist, release and song, etc. We show the usefulness of our proposed methods by comparing MusicMash2 against using some folksonomy based system directly. To the best of our knowledge, this is the first published application which shows both (i) that expressive ontology languages like OWL DL can be useful for Web applications and (ii) that querying service for OWL DL can be scalable enough to support such applications.

2 Preliminary

2.1 Folksonomies

Informally speaking, a folksonomy describes users, resources and tags in such a way that users assign tags to resources [10]. Here we follow Mika's [13] and Hotho et al.'s [10] formal model of folksonomies.

Formally, a folksonomy is a tuple $\mathbf{F} = \langle \mathcal{U}, \mathcal{T}, \mathcal{R}, A, \Theta \rangle$, where $\mathcal{U}, \mathcal{T}, \mathcal{R}$ are finite sets of users, tags and Web resources, resp., $A \subseteq \mathcal{U} \times \mathcal{T} \times \mathcal{R}$ is called tag assignments, Θ is a set of tag relations; i.e., for each member \prec of Θ , $\prec \subseteq \mathcal{U} \times \mathcal{T} \times \mathcal{T}$ is a tag relation. As Mika [13] indicated, tag relations are often thesaurus relations, such as those specified in the SKOS specification⁵. In this paper, we consider tag relations such as 'narrower' \prec_n , 'broader' \prec_b and 'related' \prec_r . For example, the following assignment $\langle u1, \text{"cat"}, r1 \rangle$ says user $u1$ uses "cat" to tag the Web resource $r1$; the following tag relation $\langle \text{"cat"} \prec_n^{u1} \text{"animal"} \rangle$ says that the tag "cat" is narrower than the tag "animal" for user $u1$. There are some properties of the above three tag relations:

1. If $\langle tag1 \prec_n^u tag2 \rangle$, then $\langle tag2 \prec_b^u tag1 \rangle$, and vice versa.
2. If $\langle tag1 \prec_r^u tag2 \rangle$, then $\langle tag2 \prec_r^u tag1 \rangle$.

It should also be noted that here we simply present a formal model of tag relations. How to extract such relations from folksonomy systems goes beyond the scope of the paper.

2.2 Ontologies and Their Semantic Approximations

As a 'representation of a shared conceptualisation' of a specific domain [7,21], ontology is regarded as a key component of the Semantic Web. Practically, ontologies are used to

⁵ <http://www.w3.org/TR/swbp-skos-core-spec/>

capture the intended meaning of a set of shared and common vocabulary of a specific domain, including important concepts (such as Artist, Album and Song), properties (such as hasAlbum and hasSong) and individuals (objects). Recently, W3C standardised the OWL Web Ontology Language [2] in its Semantic Web Activity.

It has been argued that the Web needs lightweight ontologies [13]. Recently, Calvanese et al proposed the OWL2 QL (also called DL-Lite) [5] ontology language which can express most features in UML class diagrams but still has a low reasoning overhead (worst case polynomial time, compared to worst case exponential time in the case of most widely used Description Logics-based ontology languages). A OWL2 QL ontology (\mathcal{O}) is a set of axioms of the following forms:

1. concept inclusion axioms: $B \sqsubseteq C$ where B is a basic concept $B := A \mid \exists R \mid \exists R^{-}$ and C is a general concept $C := B \mid \neg B \mid C_1 \sqcap C_2$ (where A denotes an named concept and R denotes a named property);
2. property inclusion axioms: $R_1 \sqsubseteq R_2$, where R_1, R_2 are named properties;
3. individual axioms: $A(a), R(a, b)$ where a and b are named individuals.

Straccia later proposed fuzzy DL-Lite [19]. Pan et al. [14] propose a large family of fuzzy query languages and corresponding query answering algorithms for fuzzy OWL2 QL. The TrOWL⁶ ontology reasoning infrastructure consists an ontology repository, where users can submit and share ontologies and some scalable ontology query engines, including Quill for OWL2 QL and a fuzzy OWL2 QL, supporting a large family of fuzzy queries proposed by Pan et al. [14]. Preliminary evaluations [14] have shown that Quill operates efficiently and scalably, being able to handle millions of individuals.

Pan and Thomas [15] propose a general approach to semantically approximating of ontologies, i.e., to approximate a source ontology in an expressive ontology language (such as OWL DL) into its (lowest) upper bound in a lightweight ontology language (such as OWL2 QL). This process allows conjunctive querying against OWL DL ontologies, after some careful query rewriting, to be performed within a database, while still giving soundness guaranteed results for all queries, and sound and complete (w.r.t. the source ontology) results for database-style queries. Preliminary evaluation [15] shows that the semantic approximation based OWL DL query engine in TrOWL can handle millions of individuals.

3 Motivations and Use Cases

3.1 Limitations of Folksonomy Searching

Let us first revisit the formal model of folksonomy to illustrate some limitations of tagging from the aspect of search. Given a folksonomy based system \mathcal{S} that uses the folksonomy $\mathbf{F} = \langle \mathcal{U}, \mathcal{T}, \mathcal{R}, A, \Theta \rangle$, there are three main limitations of tagging regarding search in \mathcal{S} [17]:

- Tag variation (synonymy). Consider the following two assignments: $\langle u_1, \text{“cat”}, r_1 \rangle, \langle u_2, \text{“kitty”}, r_2 \rangle$ (users u_1 and u_2 use “cat” and “kitty” to tag two different resources r_1 and r_2 , resp.). Even if u_1 and u_2 have the same meaning in mind for the two different tags, the system \mathcal{S} is not able to relate these tags in a general sense.

⁶ <http://trowl.eu/>

- Tag ambiguity (polysemy). Consider the following two assignments: $\langle u1, \text{“ma”}, r3 \rangle$, $\langle u2, \text{“ma”}, r4 \rangle$ (users $u1$ and $u2$ use “ma” to tag two different resources $r3$ and $r4$). Even if $u1$ and $u2$ use “ma” to express different ideas, the system \mathcal{S} would still return both $r3$ and $r4$ when another user (who might have a totally different idea of “ma”) searches for “ma”.
- Flat organisation of tags. Consider the following two assignments: $\langle u1, \text{“cat”}, r5 \rangle$, $\langle u2, \text{“animal”}, r6 \rangle$ (users $u1$ and $u2$ use “cat” and “animal” tag two different resources $r5$ and $r6$, resp.). When another user searches for “animal”, the system \mathcal{S} is able to return $r6$, but not $r5$.

3.2 Why Can Ontologies Help

Here are two observations of the above three limitations:

1. The issues of tag variation and flat organisation of tags indicate that the tag relations in Θ are not useful enough, because these relations are with respect to some user (or user group) only, rather than in the system level (all users).
2. The issue of ambiguity indicates we need to be able to associate tags with some contexts, so as to reduce the ambiguity.

Observation 2 clearly suggests ontology could be helpful, as they contain domain-specific knowledge. Observation 1 suggests that ontological structures should (at least) be used in the system level. Now the question is: *How can ontologies help?* In the next sub-section, we will illustrate this with some use cases.

3.3 Use Cases

This section outlines some prominent examples of ambiguity in tagging systems and how our Folksonomy Search Expansion approach could be applied in each case. The overall goal in each use case is to increase the precision of the search results. These use cases assume that an appropriate domain ontology is available in order to identify the correct individuals for expansion. The **expansion(s)** identified in each use case are shown in Table 1.

Music Videos

- *Example Search Term(s)*: “Focus”
- *Description*: A user wants to search YouTube for videos relating to a specific music artist. *Focus* is an ambiguous search term on YouTube and does not commonly relate to the artist Focus. However, YouTube users typically tag videos with both the artist name and song title. By experimentation, it can be shown that using the combination of artist name and song title as the search terms is sufficient to increase precision significantly (see Section 6.1).
- *Example Expansion*: The expansion concatenates the original search term *Focus* with the title of each song created by Focus, resulting in a set of searches.
- *Example Output*: { “Focus Hocus Pocus”, “Focus Silvia”, ... }.
- **Expansion**: Individual Property Value Expansion (IPVE) with the property specified as *createdSong* (see Table 1).

Table 1. Outline of Expansion Methods (where i represents individuals related to the search keyword(s))

Name	Description
ICE	The search terms are expanded by concatenating keywords relating to any class C , such that $C(i)$ entailed by the target ontology.
IPE	The search terms are expanded by concatenating keywords relating to any property P , such that the relation $P(i, _)$ entailed by the target ontology.
IPVE(P)	The search terms are expanded by concatenating keywords relating to the individual q , using the specified property P such that the relation $P(i, q)$ entailed by the target ontology.

Music Images

- *Example Search Term(s)*: “Metallica”
- *Description*: A user wants to search Flickr for images relating to a specific music artist. However, the search term *Metallica* returns many unrelated results ⁷. Adding further search terms increases the number of relevant results returned. However, we cannot reuse the same expansion from the previous use case, since users do not typically tag images with particular song names. In this example, the names of band members and music-related keywords (e.g., Live, Concert, Music etc.) are added to increase the search precision. Band member names would typically be found in the ABox of the ontology and music-related keyword would typically be found in the TBox.
- *Example Expansion (Band Members)*: The expansion concatenates the original search term *Metallica* with the name of each band member of Metallica.
- *Example Output (Band Members)*: { “Metallica James Hetfield”, “Metallica Lars Ulrich”, ... }.
- **Expansion (Band Members)**: Individual Property Value Expansion (IPVE) with the property specified as *hasBandMember*.
- *Example Expansion (Music-related Keywords I)*: The expansion concatenates the original search term *Metallica* with keywords related to the artist class in the ontology. Note that this particular expansion depends on the quality of annotations in the target ontology. In this example we assume that *Metallica* is an instance of classes such as *LiveBand*, *MusicArtist* etc.
- *Example Output (Music-related Keywords I)*: { “Metallica Live”, “Metallica Concert”, “Metallica Music”, ... }.
- **Expansion (Music-related Keywords I)**: Individual Class Expansion (ICE)
- *Example Expansion (Music-related Keywords II)*: Depending on the structure of the target ontology, the expansion can also be performed by concatenating the original search term *Metallica* with the keywords related to the individual’s properties. For example, the property *performedLiveAt* contains the keywords ‘performed’ and ‘live’.
- *Example Output (Music-related Keywords II)*: { “Metallica Live”, “Metallica Performed”, ... }.
- **Expansion (Music-related Keywords II)**: Individual Property Expansion (IPE).

⁷ Flickr’s tag search returned 3 out of 25 relevant results (12-Dec-2008).

Film Clips

- *Example Search Term(s)*: “The Wicker Man”
- *Description*: A user wants to search YouTube for videos clips related to the 1973 film *The Wicker Man*. However, *The Wicker Man* could refer to the Iron Maiden song *The Wicker Man* or the 2006 remake of the original film. Adding some context to the original search terms can help disambiguate between the various different meanings of *The Wicker Man*. In this example, including the names of associated people such as the director, writer and starring actors increases the relevance of the search results.
- *Example Expansion (associated people)*: The expansion concatenates the original search terms *The Wicker Man* with the names of starring actors, writer and director, resulting in a set of searches.
- *Example Output*: {“The Wicker Man Edward Woodward”, “The Wicker Man Christopher Lee”, “The Wicker Man Anthony Shaffer”, ...}.
- **Expansion**: Individual Property Value Expansion (IPVE) with the property specified as *hasAssociatedPerson*, such that in the target ontology *hasAssociatedPerson* is a super-property of the *hasActor*, *hasDirector* and *hasWriter* properties.

Table 1 outlines the three general Folksonomy Search Expansion methods motivated by the use cases. Each expansion method firstly identifies individuals i that relate to the original search keyword(s) provided by the user. The output from each expansion method is a set of expanded search terms constructed by concatenating each result of the expansion to the original search term. In the next section we will formally define the proposed expansions.

4 Folksonomy Search Expansion

The section presents the details of our Folksonomy Search Expansion approach to address the problem of ambiguity in tagging systems. Firstly we show how keywords are associated with ontologies. We then present a general framework of ontology-based folksonomy search expansion methods based on the use cases in Section 3.3.

4.1 Automatic Keyword-Ontology Association

Keywords appear in annotations and datatype properties of ontologies. These keywords can be associated with concepts, properties and individuals (in ontologies), with scores n based on ranking factors⁸. These scores in turn can be used to calculate the *tf·idf* [18] for each keyword, and normalise them using a sigmoid function such as the one shown in (1) to a degree between 0 and 1.

$$w(n) = \frac{2}{1.2^{-n} + 1} - 1 \quad (1)$$

Formally, for each ontological element i (concept, property, object) in a given ontology \mathcal{O} , there exists a set of associated keywords $K(i)$ and each keyword $k \in K(i)$ associates to i with a degree $d(i, k) \in (0, 1]$. Similarly, each keyword k is associated with a set of classes $C(k)$, properties $P(k)$ and individuals $O(k)$.

⁸ <http://www.seomoz.org/article/search-ranking-factors>

4.2 Ontology-Based Folksonomy Search Expansions

Informally, the main ideas of ontology-based folksonomy search expansion is to make use of the (implicit or explicit) structure of ontologies to relate keywords with other keywords. Accordingly, a search expansion $E(k, expansionMethod)$ is a function that takes a tag (keyword) k and the expansion method as inputs and returns a set of keywords. Now we formally define the search expansions identified in Section 3.3.

Individual Property Value Expansion (IPVE) returns the set of associated keywords of all individuals that relate (via a given property P) to all the individuals in $O(k)$.

$$E(k, IPVE(P)) = \bigcup_{\forall i \in O(k)} \bigcup_{\forall q: \mathcal{O} \models P(i, q)} K(q).$$

Individual Class Expansion (ICE) returns the set of associated keywords of all classes of all the individuals in $O(k)$.

$$E(k, ICE) = \bigcup_{\forall i \in O(k)} \bigcup_{\forall C: \mathcal{O} \models C(i)} K(C).$$

Individual Property Expansion (IPE) returns the set of associated keywords of all properties that relate to all the individuals in $O(k)$.

$$E(k, IPE) = \bigcup_{\forall i \in O(k)} \bigcup_{\forall P: \mathcal{O} \models P(i, _)} K(P).$$

As we consider entailments in the given ontology \mathcal{O} , the computational cost could be expensive, since entailment problem in OWL DL is NEXPTIME-complete. It is important to point out that, all the required ontology entailments in the above expansion methods are included in the DL-Lite semantic approximation of a given OWL DL ontology. This suggests that theoretically, by making use of semantic approximations, the proposed methods can be scalable. This is confirmed by our evaluation presented in Section 6. Furthermore, if the set of tags returned by search expansions is too large, one could apply some strategies to keep it small, such as by setting a threshold for the association degrees. A typical scenario where our approach could be used is in domain-specific mashup application. In such a system the developer should tailor the exact expansion used to their own ontology, application domain and target tagging system.

5 Implementations

This section describes how the proposed approach is implemented in our ontology infrastructure and illustrates how to make use of the ontology infrastructure in a semantic mashup application.

5.1 Ontology Infrastructure

The proposed automatic keyword-ontology association and the proposed ontology-based folksonomy search expansion methods are implemented in TrOWL the ontology reasoning infrastructure (which is made use of by ONTOSEARCH2 [16,20] the

ontological search engine for entailment-plus-keyword search) and Taggr the ontology-based folksonomy search optimiser, respectively.

A useful feature of the TrOWL repository is that it automatically associates keywords (in values of annotation properties as well as implicit metadata in target ontologies) with concepts, properties and individuals in the ontologies. Default annotation properties include the `rdfs:label`, `rdfs:comment`, `rdfs:seeAlso`; we also define the `dc:title`, `dc:description`, and `foaf:name` properties (from Dublin Core⁹ and FOAF¹⁰) as annotation properties. Implicit metadata is drawn from the namespace and ID of each artifact in the ontology. When users upload their ontologies, TrOWL computes the semantic approximation of the ontologies as described in [15] and computes keyword-ontology association as presented earlier in the paper.

Taggr provides an ontology-enabled common interface for folksonomy based systems¹¹. It stores a basic ontology (which we refer to as the “tagging database”) in TrOWL, capturing the relationships between users, tags and resources in the folksonomy based systems it supports. It provides the functionality of gathering resources and their related tags from the systems that it supports, and then populate them to its tagging database from time to time. Furthermore, Taggr allows users to specify which search expansion method(s) and which reference ontology(-ies) to use for the expansion. The extended searches are evaluated against its tagging database. As the ontological constraints needed for the proposed search expansion require only the expressive power of OWL2 QL, Taggr can make use of the semantic approximation(s) of the reference OWL DL ontology(-ies) for all the entailment checking. Due to the logical properties of semantic approximation, TrOWL can provide sound and complete results for all the needed entailment checking. As mentioned in Section 4, all the proposed search expansion methods are available for application developers to choose, which allows search expansions to be tailored to the nature of tagging in different domains and types of resources that are to be retrieved.



Fig. 1. MusicMash2 Alpha: <http://www.musicmash.org/>

⁹ <http://dublincore.org/>

¹⁰ <http://www.foaf-project.org/>

¹¹ Taggr currently supports YouTube and Flickr.

5.2 Case Study: MusicMash2

MusicMash2 is a semantic mashup application (see Figure 1), which is intended to integrate music-related content from various folksonomy based tagging systems and music meta-data Web services.

As discussed in Section 3.1, search both within and across folksonomy based systems is an open problem. A naive approach to folksonomy search, such as those provided by most tagging systems¹², results in unacceptable precision in domain specific searches (see Section 6.1). MusicMash2 addresses this problem by making use of the Folksonomy Search Expansion methods provided by Taggr. More specifically, since the music content of interest to MusicMash2 can be naturally expressed as a populated domain ontology. Currently MusicMash2 addresses the problem of ambiguity when search for music videos related to music artists, albums and tracks. Instance Property Value Expansion (IPVE) is chosen as the search expansion method for MusicMash2. For example, when searching for videos relating to a specific artist, MusicMash2 will call Taggr to make the following expansion, $E(\text{artistName}, \text{IPVE}(\text{foaf:made}))$, i.e., to search with some artist name (*artistName*) for everything related to it via the foaf:made property. The goal of this expansion is to reduce the ambiguity of the original search term *artistName* by providing a set of *artistName*, *songTitle* keyword sets.

To use the IPVE search expansion method, developers of MusicMash2 need to have a domain ontology on music, populated with data needed by MusicMash2. Firstly, the Music Ontology¹³ is chosen here as it provides the main classes and properties for describing music on the Web. Secondly, to populate the Music Ontology, MusicMash2 makes use of Web services which provide needed information (each in their own proprietary format), such as, MusicBrainz¹⁴, Last.fm¹⁵ and DBpedia¹⁶. MusicMash2 maps the data from each music meta-data Web service to the standardised Music Ontology format and submits the populated ontology to ONTOSEARCH2. Such populated ontology is exploited by the IPVE search expansion method. Since it is stored in ONTOSEARCH2, the populated ontology can also be reused by users of ONTOSEARCH2.

6 Evaluation

In this section, we present our evaluations (on usefulness and scalability) of Individual Property Value Expansion (IPVE) to address the problem of ambiguity illustrated by the Music Video use case in Section 3.3. We have used our case study application, MusicMash2, to provide a basis to refine the implementation of this particular expansion method. For this reason, we believe that an evaluation of IPVE provides a good practical illustration of how a Folksonomy Search Expansion method can perform in a real-world application.

¹² YouTube Developer API: <http://www.youtube.com/dev/>

¹³ <http://www.musicontology.com/>

¹⁴ <http://www.musicbrainz.org/>

¹⁵ <http://www.last.fm/>

¹⁶ <http://dbpedia.org/>

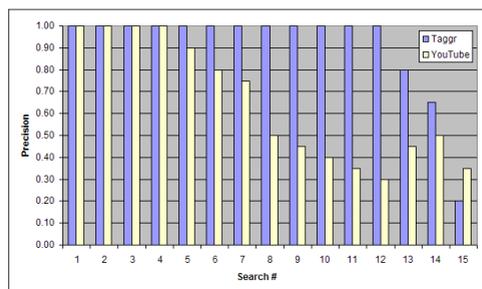


Fig. 2. Result set precision for *top k* search

6.1 Usefulness of Search Expansions

Based on our case study (see Section 5.2), we evaluated the usefulness of our approach by performing identical *top k* searches for music artists with YouTube and Taggr using the IPVE search expansion. We provide this evaluation to show how our system deals with ambiguity in the music domain, which we feel is a typical application of our Folksonomy Search Expansion approach. The set of 15 search terms (see Table 2) used in the evaluation have been chosen to evaluate how well Taggr with IPVE search expansion handles ambiguity in the music domain when compared with the standard YouTube search facilities. We have also included some less ambiguous artist names such as Radiohead and Metallica for comparison.

First of all, we performed *top k* searches using each system to retrieve the top 20 results. We then classified each result as either relevant or irrelevant with respect to the meaning of the original search. In the case of this evaluation, a result was classified as relevant if it featured the specified artist. The precision of each result set (see Figure 2) was calculated using the function (2); where n_{tp} is the number of relevant results and n_{fp} is the number of irrelevant results in the result set.

$$P(n_{tp}, n_{fp}) = \frac{n_{tp}}{n_{tp} + n_{fp}} \quad (2)$$

To prepare for this evaluation, Taggr’s tagging database was populated with the top 500 results from YouTube for each of the search terms in Table 2. These 500 results typically included both music and non-music related content. Taggr IPVE made use of a populated Music Ontology for search expansion, with the role specified as foaf:made. More precisely, the expansion $E(\text{artistName}, IPVE(\text{foaf:made}))$ was used, where *artistName* is the full name of the artist used as the search term.

Fig. 2 shows that in all but one search, Taggr outperformed YouTube in terms of precision. YouTube returned good results for unambiguous artist names but poor results for ambiguous names. In searches #1 - 4 both YouTube and Taggr return results with perfect precision with these unambiguous artist names. With the more ambiguous artist names in searches #5 - 12 YouTube performs worse, while Taggr retains perfect precision. In searches #13 and #14, Taggr loses some precision for the highly ambiguous artist names “Yes” and “Kiss”. In the final search, #15, Taggr performs slightly worse

than YouTube, by 15%. Our initial reaction is that for highly ambiguous artist names and song titles, the artist name plus song title combination used by IPVE is less effective than YouTube’s built in search facilities. A simple approach to addressing this issue would be to attempt to rank the results by examining the complete set of tags for a resource and checking how many are actually matched by the keywords generated in the search expansion.

This evaluation is intended to show how our approach performs in terms of reducing ambiguity in a real-word system. We have chosen the IPVE search expansion to illustrate the approach, since this particular expansion method seems to be most widely applicable (see Section 3.3).

Table 2. Artist names used in the evaluation

Search #	Search Term	Search #	Search Term
1	The Beatles	9	Thursday
2	Porcupine Tree	10	Pelican
3	Radiohead	11	Focus
4	Metallica	12	Isis
5	Eels	13	Yes
6	Doves	14	Kiss
7	Finch	15	Jet
8	Strung Out		

6.2 Scalability of Search Expansions

First of all, it should be noted that the scalability of this application depends on the performance of TrOWL. Several evaluations [15,14] of TrOWL have been made using the Lehigh University Benchmark [8], and these have shown that TrOWL is scalable for large ontologies, containing of millions of individuals.

We have evaluated the performance of IPVE in Taggr by using two different sized datasets. More specifically, Taggr was used to perform the following search expansion $E(\text{artistName}, IPVE(\text{foaf:made}))$, i.e., search with some artist name (*artistName*) for everything related to it via the foaf:made property. The evaluation was carried out using a typical desktop computer (see Table 3).

This evaluation relies on a populated Music Ontology being present in ONTOSEARCH2. MusicMash2 was used to prepare the data for the evaluation. A list of artists was created by selecting the most popular artists from the Audioscrobbler web services (using the example user profile from the API documentation¹⁷). The list was then expanded by including the top 50 similar artists for each of the artists. MusicMash2 then performed a batch process to retrieve information relating to each artist (including other artists) from MusicBrainz¹⁸, DBpedia¹⁹ and Audioscrobbler²⁰ web services. This information was used to populate the two versions of the Music Ontology.

¹⁷ <http://ws.audioscrobbler.com/1.0/user/RJ/topartists.xml>

¹⁸ <http://www.musicbrainz.org/>

¹⁹ <http://www.dbpedia.org/>

²⁰ <http://ws.audioscrobbler.com/>

Table 3. Specifications of test computer

Operating System	Windows XP Pro
CPU	Intel Pentium 4 3.0GHz Dual Core
RAM	2Gb
Hard Disk	80Gb SATA
JVM	Sun JVM version 1.6
Database	PostgreSQL 8.2

Table 4. Individual Property Value Expansion (IPVE) Performance

$N_{Artists}$	N_{Albums}	N_{Songs}	N_{Axioms}	T_{IPVE} (ms)
1303	2,531	30,182	~1,000,000	155
5521	11,336	136,040	~4,500,000	2036

Table 4 shows the performance of the system working with a small dataset of 1,300 artists (30,000 total individuals) and a larger dataset of 5,500 artists (136,000 total individuals). The larger dataset results in an ontology containing over 4,500,000 ground axioms. It is clear to see that further evaluations with larger datasets will need to be carried out to determine how this approach scales with extremely large datasets. Given the time (about 2 seconds for the larger dataset, see the T_{IPVE} column in Table 4) used for the IPVE search expansion, we believe that with a dedicated database server and further optimisations of our prototype implementation, this application will scale effectively over ontologies containing billions of axioms.

7 Discussion and Outlook

How to apply Semantic Web technologies to improve folksonomy based systems and social networks has been a pressing issue for the Semantic Web community [13]. In this paper, we have addressed the problem of ambiguity in tagging systems with a general framework of Folksonomy Search Expansion methods, based on ontology entailments, showing that different expansion methods can be useful in potentially different situations. In particular, we use a semantic mashup application, i.e. MusicMash2, to demonstrate the use of the IPVE search expansion method and evaluate its usefulness and scalability. The results of our preliminary evaluation indicate that the proposed technique can handle very large ontologies and allows folksonomies to be searched more effectively, returning more relevant results for folksonomy search than the keyword based approach.

By using our ontology infrastructure to expand folksonomy search transparently for users, we have solved the open problem of how to exploit the benefits of ontology to improve folksonomy search, without bothering untrained users with its rigidity. This means that users, such as those of MusicMash2, can exploit the benefits of ontology enhanced folksonomy search without the knowledge of existing of neither the ontology infrastructure nor the related ontologies. Interestingly, for all the Folksonomy Search Expansion methods, the required ontology entailments are within the expressive

power of OWL2 QL, which allows us to make use of OWL2 QL semantic approximations [15] of given domain ontologies. This means that, although the domain ontologies can be very complicated, the Folksonomy Search Expansion methods only require the OWL2 QL approximations of such ontologies. It has been argued that the Web needs lightweight ontologies [13]. Our work suggests that the Web needs lightweight representation/approximations of (potentially complicated but still useful) ontologies.

While this paper addresses the problem of ambiguity in tagging systems, we believe that our Folksonomy Search Expansion approach can be extended to address the lack of structure and tag variation limitations. For example, users searching for a high level concept such as *animal* may find that resources in tagging systems are not frequently tagged with such a high level concept. To address the problem of search across tagging systems, different expansions could be used for each target tagging system. For example, in Section 3.3 we use different expansions to search YouTube and Flickr for resources related to music artists. We are currently working on implementing and evaluating these extensions to our approach and expect that they should have similar benefits to our ambiguity approach presented in this paper.

References

1. Ankolekar, A., Krotzsch, M., Tran, T., Vrandečić, D.: The two cultures: mashing up web 2.0 and the semantic web. In: Proc. of the 16th international World Wide Web conference (WWW 2007), pp. 825–834 (2007)
2. Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D.H.L., Patel-Schneider, P.F., Stein, L.A. (eds.): OWL Web Ontology Language Reference, February (2004), <http://www.w3.org/TR/owl-ref/>
3. Benjamins, R.V., Davies, J., Baeza-Yates, R., Mika, P., Zaragoza, H., Greaves, M., Gomez-Perez, J.M., Contreras, J., Domingue, J., Fensel, D.: Near-term prospects for semantic technologies. *IEEE Intelligent Systems* 23(1), 76–88 (2008)
4. Bhogal, J., Macfarlane, A., Smith, P.: A review of ontology based query expansion. *Information Processing & Management* (4), 866–886 (2007)
5. Calvanese, D., De Giacomo, G., Lenzerini, M., Rosati, R., Vetere, G.: DL-Lite: Practical Reasoning for Rich DLs. In: Proc. of the DL 2004 Workshop (2004)
6. Greaves, M.: Semantic web 2.0. *IEEE Intelligent Systems* 22(2), 94–96 (2007)
7. Gruber, T.R.: Towards Principles for the Design of Ontologies Used for Knowledge Sharing. In: Guarino, N., Poli, R. (eds.) *Formal Ontology in Conceptual Analysis and Knowledge Representation*, Dordrecht, The Netherlands. Kluwer Academic Publishers, Dordrecht (1993)
8. Guo, Y., Pan, Z., Heflin, J.: LUBM: A Benchmark for OWL Knowledge Base Systems. *Journal of Web Semantics* 3(2), 158–182 (2005)
9. Halpin, H., Robu, V.n., Shepherd, H.: The complex dynamics of collaborative tagging. In: Proc. of the 16th International World Wide Web Conference (WWW 2007), pp. 211–220 (2007)
10. Hotho, A., Jäschke, R., Schmitz, C., Stumme, G.: Information retrieval in folksonomies: Search and ranking. In: Sure, Y., Domingue, J. (eds.) *ESWC 2006*. LNCS, vol. 4011, pp. 411–426. Springer, Heidelberg (2006)
11. Jackson, C., Wang, H.J.: Subspace: secure cross-domain communication for web mashups. In: Proc. of 16th International World Wide Web Conference (WWW 2007), pp. 611–620 (2007)
12. Lassila, O., Hendler, J.: Embracing "web 3.0". *IEEE Internet Computing* 11(3), 90–93 (2007)

13. Mika, P.: Ontologies are us: A unified model of social networks and semantics. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 522–536. Springer, Heidelberg (2005)
14. Pan, J.Z., Stamou, G., Stoilos, G., Taylor, S., Thomas, E.: Scalable Querying Services over Fuzzy Ontologies. In: Proc. of the 17th International World Wide Web Conference (WWW 2008) (2008)
15. Pan, J.Z., Thomas, E.: Approximating OWL-DL Ontologies. In: Proceedings of the 22nd Conference on Artificial Intelligence (AAAI 2007), pp. 1434–1439 (2007)
16. Pan, J.Z., Thomas, E., Sleeman, D.: ONTOSEARCH2: Searching and Querying Web Ontologies. In: Proc. of WWW/Internet 2006, pp. 211–218 (2006)
17. Passant, A.: Using Ontologies to Strengthen Folksonomies and Enrich Information Retrieval in Weblogs. In: Proc. of 2007 International Conference on Weblogs and Social Media (ICWSM 2007) (2007)
18. Salton, G., McGill, M.J.: Introduction to modern information retrieval. McGraw-Hill, New York (1983)
19. Straccia, U.: Answering vague queries in fuzzy DL-Lite. In: Proc. of the 11th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2006), pp. 2238–2245 (2006)
20. Thomas, E., Pan, J.Z., Sleeman, D.: ONTOSEARCH2: Searching Ontologies Semantically. In: Proc. of OWL: Experiences and Directions (OWL-ED 2007) (2007)
21. Uschold, M., Gruninger, M.: Ontologies: Principles, Methods and Applications. The Knowledge Engineering Review (1996)