

# Exploiting Semantic Web Techniques for Representing and Utilising Folksonomies

Owen Sacco

LUSSI Department, Telecom Bretagne  
Technopôle Brest-Iroise – CS 29238  
Brest Cedex 3 - France

owensacco@gmail.com

Cécile Bothorel

LUSSI Department, Telecom Bretagne  
Technopôle Brest-Iroise – CS 29238  
Brest Cedex 3 - France

cecile.bothorel@telecom-bretagne.eu

## ABSTRACT

In this paper, an ontology is proposed to represent hierarchical structure of tags unfolded using the “fast unfolding of communities in large networks” [19] algorithm. This paper also proposes a semantic state of the art application to transform communities of tags computed by this algorithm into a semantic format in light of enhancing search and exploration for Web content.

## Categories and Subject Descriptors

D.3.2 [Programming Languages]: Language Classifications – Java, I.2.4 [Artificial Intelligence]: Knowledge Representation Formalisms and Methods – *Representation languages*.

## General Terms

Languages, Theory.

## Keywords

RDF, OWL, SCOT, Semantic Web, Jena, Corese, Community Structure

## 1. INTRODUCTION

The traditional World Wide Web (WWW), currently referred to as Web 1.0, lacked user collaboration and involvement due to the static nature of how content was structured. This static structure disallowed features for content sharing, automatic knowledge acquisition and interoperability amongst Websites. As websites grew in number, together with an increase in Web users, the need for more user involvement increased that motivated scientists to enhance Web technologies. Once these Web technologies started to mature, the vision of the Semantic Web was defined. Moreover, as Web technologies improved, the Web evolved into a second generation known as Web 2.0 that brought an increase in user involvement on the Web. Web 2.0 is considered more of a Social Web since people interact more and are more connected with each other. With the introduction of these enhanced technologies, Web 2.0 websites started to sprout

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MSM'10, June 13, 2010, Toronto, Canada.

Copyright 2010 ACM 978-1-4503-0229-6/06/10...\$10.00

with improved user-friendly features. One of these features is known as social tagging. This consists of features that allow users to annotate content, such as pictures, video, bookmarks, blogs and any other content with keywords known as tags. Despite the Web experienced a growth in the number of tags, tags are still in a format that lack meaning for machines to infer more knowledge from such tags. In this research, current Web techniques for enriching tags on the Web are discussed, together with current Web technologies that are available for semantically describing resources on the Web. Moreover, in this research paper, a state of the art application is proposed that transforms textual tags representation into a semantic representation. The tags used in this research have been gathered from a social bookmarking website called Bibsonomy<sup>1</sup> and the relationships amongst such tags are unfolded by using graph mining techniques. The aim of this application is to promote the importance of representing tags in Semantic Web formats to improve searching and exploration of content through the use of tags.

## 2. SEMANTIC WEB

The Semantic Web provides new approaches to managing information and processes on the Web. This is achieved by using metadata to describe Web data and also Web services. The advantages of using metadata are that information can be organised better and also information can be found on the basis of the content's meaning rather than processing content merely as text. This implies that when searching for specific terms, contents can be distinguished especially when words can have different meanings. Additionally, when a particular subject of a search term cannot be searched, other related subjects can instead be induced from metadata. Another advantage is that the semantics in metadata improved the way information is presented. Information can be represented in clusters according to the information they relate to. In addition to presenting information, semantic data can allow information to be merged from heterogeneous sources on the basis of the relationships amongst data, even if the underlying data schemas differ.

The Semantic Web encouraged the creation of new technologies such as new meta-formats. These meta-formats represent metadata in a format that can be processed by machines to infer additional information, to allow for data sharing and to allow for interoperability amongst Web pages. The common format and recommended by W3C for semantic data representation is the

<sup>1</sup> Bibsonomy: <http://www.bibsonomy.org/>

Resource Description Framework (RDF). RDF consists of a data model to describe data on the Web. The RDF model can also be queried by using an RDF query engine called SPARQL. Moreover, the RDF model, in some instances may require more meaning to describe its structure. Therefore, an RDF vocabulary can be used to describe the RDF model's structure. This vocabulary is called the RDF Schema (RDFS). Apart from vocabularies, an RDF model may contain data that pertain to a specific domain which such domain's structure needs to be explicitly defined. A Semantic Web technology known as Ontology describes domain specific structures. These Semantic Web technologies, namely: RDF, SPARQL, RDFS and Ontology are described below.

## 2.1 Resource Description Framework (RDF)

RDF is a framework that describes resources on the World Wide Web. Resources can be anything that can be described on the Web, even when the resource cannot be directly retrieved. RDF provides a framework for representing data that can be exchanged without loss of meaning. RDF is suitable for merging data over the Web even if the underlying data schemas are different between one another. RDF uniquely identifies resources on the Web by means of Internationalised Resource Identifiers (IRIs), formerly known as Uniform Resource Identifiers (URIs). Resources are described in RDF in the form of triple statements. A triple statement consists of a Subject, a Predicate and an Object. A subject consists of the unique identifier that identifies the resource. A predicate represents the property characteristics of the subject that the resource specifies. An object consists of the property value of that statement. Values can be either literals or other resources. Therefore, the predicate of the RDF statement describes relationships between the subject and the object.

The RDF model can be serialized into various formats such as N-Triples, RDF/JSON, RDF/XML and Turtle. The widely used and recommended serialised format is the RDF/XML. This format represents the RDF graph in an XML based syntax which is convenient for data exchange. Moreover, the advantage of describing RDF models in XML is that it provides a format that is already accessible by machines through XML parsers. However, since XML does not contain any structure to define any semantics of the data or even the relationship between data, RDF adds value to XML by adding semantics and relationships to data.

## 2.2 SPARQL Protocol and RDF Query Language

SPARQL is considered as the W3C recommended query language to query RDF models. SPARQL contains a similar syntax to SQL, which is the widely used query language for querying relational databases. The reason behind having a SQL like syntax is, since most data on the web is stored in relational databases and most websites are developed by querying such relational databases, developers familiar with SQL do not need to learn a new language. This encourages users or developers to easily adapt to using such powerful web technologies and also it would not be cumbersome to integrate these technologies with existing data. SPARQL queries take the form of a set of triple patterns called a basic graph pattern. SPARQL triple patterns are

similar to RDF triples with the difference that SPARQL triples, each subject, predicate and object can be bound to a variable. When the subject, predicate or object are bound to a variable, after executing a SPARQL query on an RDF model, the bounded variables are mapped to the query results. SPARQL queries consist mainly of two parts: the `SELECT` clause that identifies the variables to appear in the query result, and the `WHERE` clause that provides the basic graph pattern to match against the data graph. The SPARQL results can be represented in SPARQL endpoints which although conforms to legal RDF structures, the results are structured according to the bounded variables. The SPARQL results can also be represented in RDF statements by using the `CONSTRUCT` clause instead of the `SELECT` clause.

## 2.3 Resource Description Framework Schema (RDFS)

RDF vocabularies are essential since RDF is a general-purpose language for representing information on the Web, and does not describe the meaning of specific classes and properties. RDF vocabularies describe classes and properties using the RDF Vocabulary Description Language, known as RDF Schema (RDFS), which indicate what an RDF statement is about. Explicitly, RDFS indicates that the statements are describing particular types of classes of resources that contain specific kinds of properties. RDFS does not provide a vocabulary containing specific purpose classes such as "Book", nor specific purpose properties such as "Title". Instead, RDFS provides a structured vocabulary that can be used to describe such classes and properties, and also to state what properties will be used to describe the classes. Therefore, RDFS provides a type system for RDF that statements in RDF documents can refer to and it provides additional information to the resources described in the RDF documents. This type system is used to validate the value in RDF documents and also to check any restrictions that might be implied on properties within RDF documents. In other words, the RDF Schema is used to check the integrity of the triples of RDF statements; that the predicate is proper for the subject and the object is proper for the predicate.

The RDF Schema descriptions are in the form of legal RDF graphs. Hence, RDF Schema contains specialised sets of predefined RDF resources together with their specific meanings. Even though an RDF parser might manage to parse an RDF Schema, the intended meaning of such vocabularies might not be "understood" by such parsers. This implies that RDF software must be developed to process the added meaning of RDF vocabularies.

## 2.4 Ontology

RDF Schema describes the generic structure and restrictions of RDF documents but does not provide domain specific vocabularies. Ontologies are therefore referred to as additional vocabularies which focus on specific domains. Ontology is defined as "a formal, explicit specification of a shared conceptualisation." Other authors define ontology as a "shared, explicit but partial specification of the commonly agreed upon intended meaning of a conceptualisation." In other words, parties with a common concept of data specify as clearly as possible such concepts and they can build systems on the basis of these

specifications that allow the systems to interoperate with one other. Ontology is therefore a common vocabulary and semantic interpretations of specific terms that can be processed by machines for the purpose of sharing and manipulating information. Ontologies can either be expressed in RDF, or in RDFS, or most commonly expressed using the ontology language known as Web Ontology Language (OWL). OWL, now superseded by OWL 2, is a declarative language used to express ontologies in a logical way. With the means of software tools called reasoners, further information can be induced out of the logical reasoning behind the ontologies. OWL 2 documents describe ontologies by means of classes, properties, objects (known as individuals in OWL 2) and data values. OWL 2 documents are based on RDF models and therefore are machine readable documents. OWL 2 documents can be processed alongside with RDF documents, and also, it is mandatory that OWL 2 documents are exchanged as RDF/XML documents.

### 3. FOLKSONOMIES AND THEIR SEMANTIC REPRESENTATIONS

One of the most widely used feature in Web 2.0 websites is social tagging and it is currently an increasing popular activity on the social web. This social phenomenon consists of active users voluntarily annotating any resource with keywords on social websites. These key words are known as tags, this web activity is known as tagging and the user tagging the resource is known as the tagger. This act of tagging was coined by Thomas Vander Wal [2] as folksonomy, and he defined folkosnomy as “the result of personal free tagging of information and objects (anything with a URL) for one’s own retrieval. The tagging is done in a social environment (usually shared and open to others). Folksonomy is created from the act of tagging by the person consuming the information.”

Tagging is popular in social websites such as photo sharing websites (Flickr<sup>2</sup>), video sharing websites (YouTube<sup>3</sup>) and social bookmarking websites (delicious<sup>4</sup>). In order to annotate a web resource, users do not require any particular expertise and hence, tags are simple to create. Due to the simple nature of this activity, it encourages users to tag as many resources as they deem to by describing any resource with whatever word that comes into their mind at that instance. This creates flexibility on the Web since there is no strict way how to order resources and the user does not need any expertise in any subject to label resources. Hence, tags can be considered as a rich free-form classification method without adhering to any pre-determined taxonomy, or any strict classification scheme, or any controlled vocabulary. Moreover, users are allowed to tag a resource with more than one tag.

Apart from ordering and classifying resources with tags, tags are also useful for searching for resources. However, there are several issues associated with searching for resources by means of tags. Several of these problems, as highlighted in [4], include:

- Since tags are user’s perception of resources, resources can be tagged with a term that is not related with the resource. When having many resources tagged with non-related terms, this will cause noise because non-related resources will be retrieved making the search result inaccurate. The user searching for resources will have to filter out those unrelated resources by evaluating and selecting the correct resources.
- Tags can be misspelled or abbreviated or written in a way how the tagger deems fit. Additionally, different synonyms, which consist of different words that describe the same thing, for example lorry and truck, can be used as tags and thus, many related resources will be tagged with different synonyms. Many of these resources might not be retrieved since they are not tagged with the same term as the one being searched even though these resources are related to each other that might be useful for the user.
- Words have different meanings for instance apple referring to a fruit or referring to the computer company. These ambiguous words that have different meanings can yield undesired results.
- Due to the fact that users have different perceptions, users tag resources with terms that are generic for some users or that are too specific for others. These differences in level of granularity may opt out resources in the retrieved search results.

Apart from these issues, since tags are textual in nature and do not contain any additional semantics, these tags cannot be processed by machines to infer other knowledge. Therefore, it is of paramount importance that folksonomies are studied in order to find semantic representations for tags on the Web. As mentioned previously, the recommended format for representing semantic data is in RDF models and hence, tags must be described using the RDF model structure. This semantic format enriches tags with semantics that can be utilised by machines to infer other knowledge for instance to relate different resources. Moreover, to semantically enrich the tags and the tagging activity, it is optimal to have an ontology that describes the domain of folksonomies to which the metadata of tags can adhere to.

#### 3.1 Ontologies for folksonomies

This section describes several current ontologies that can be used to describe several aspects of folksonomies.

##### 3.1.1 Richard Newman’s Tag Ontology

Richard Newman [6] proposes a tag ontology to “model the relationship between an agent, an arbitrary resource, and one or more tags.” In this relation, the agent represents who tagged the resource; the resource represents the object being tagged; and the set of labels used as tags to annotate the resource. Tags are defined using a `Tag` class and the tagger is defined using the FOAF<sup>5</sup> ontology.

---

<sup>2</sup> Flickr: <http://www.flickr.com/>

<sup>3</sup> YouTube: <http://www.youtube.com/>

<sup>4</sup> Delicious: <http://delicious.com/>

---

<sup>5</sup> The Friend-Of-A-Friend (FOAF) ontology defines person-related resources on the Web.

### 3.1.2 TagOntology

The TagOntology [7] is proposed by Tom Gruber and he basis the ontology on the five-places relation principle: `Tagging(object, tag, tagger, source, + / -)`. The object in this model represents the content which is being tagged; the tag is the label or word used to tag with; the tagger represents who tagged the object; the source is the system where the actual tagging model is stored; and a polarity that represents a + or -, which is "a vote" of the tagging fact to assert that the tagging fact is true or not.

### 3.1.3 Meaning Of A Tag (MOAT)

The Meaning Of A Tag (MOAT) [8] represents a light weight ontology that illustrates how tags can be linked to different meanings by using URIs that represent resources on the Web. MOAT defines global and local meanings for each tag. Global meanings are those possible meanings that could be given to a tag. Local meanings are those meanings that describe best a tag during the tagging activity. MOAT extends the `Tag` class described in Richard Newman's Tag Ontology by describing a unique label for each tag that in Richard Newman's Tag Ontology this restriction is not accounted for. Tags are described with global meanings by a `hasMeaning` property which is contained in the `Meaning` class. For local meanings, tags are represented using the `RestrictedTagging` class combined with a `tagMeaning` property. The FOAF ontology is used to define who tagged the resource.

### 3.1.4 Social Semantic Cloud Of Tags (SCOT)

The Social Semantic Cloud Of Tags [9] provides an ontology for expressing the tagging activity by defining tag data. This ontology is also used to provide social interoperability by sharing tag data for reuse and representing relations amongst individuals across the Social Web. The main aim of SCOT is to represent the main tagging activity that involves the resources, the people (who tagged the resource) and the labels of tags. The SCOT ontology is build on top of the concept of Richard Newman's Tag Ontology. The fundamental element of the SCOT ontology is the `scot:TagCloud` RDFS class that identifies a resource by a URI. This class represents the tag cloud itself and contains relationships to link to other elements and properties that are described even in other ontologies.

### 3.1.5 CommonTag

The CommonTag ontology [10] is designed with the purpose of adding concepts to tags from databases such as Freebase<sup>6</sup> and DBPedia<sup>7</sup>. The CommonTag ontology can also be used to link tags to specific concepts that are described elsewhere other than

---

<sup>6</sup> <http://www.freebase.com/>

Freebase provides datasets built by communities that are freely accessible. Freebase offers tools that help developers access and control the content contained within these datasets.

<sup>7</sup> <http://dbpedia.org>

DBPedia extracts information from the online encyclopaedia Wikipedia and provides such information in a semantic format that can be processed by machines.

the aforementioned databases. The links between tags and the concept resources are achieved by means of URIs.

## 3.2 Current methodologies to integrate folksonomies with the Semantic Web

Despite the tags are described with semantics on the Web using the above ontologies, several issues explained previously for searching content by querying tags still need to be solved. This section explains several methodologies that try to solve issues when relating resources together for enhancing searching and exploration of tags.

One way of relating resources is by using the popular tags amongst resources. However, it is not sufficient to rely only on such popular tags. This is because tags, which are words, may have more than one meaning and non-related web resources may also be tagged with similar tags but implying different meanings. In order to distinguish resources from each other, it is therefore of paramount importance to take into consideration all the tags used to annotate a web resource as a set of tags rather than considering the tags as individual tags.

Several scholarly work [11-18] suggest to add a concept or meaning to a tag whilst tagging a web resource so that from these meanings, tags having the same concept can be related together. This suggestion involves retrieving concepts from knowledge data stores such as DBpedia, GeoNames, Wikipedia or WordNet. In some of these works it was proposed that when adding concepts to tags, this involves providing the user with several concepts for that word and the user must select the best meaning for the tag used to describe the web resource. This non-automatic procedure relies on the user to match the tag to its proper meaning and this adds an additional step to the tagging activity. This additional step might be cumbersome for the user in a way that it might cause the tagging activity to reduce its popularity. Moreover, the user might select an incorrect concept for the intended meaning of the tag that will defeat the scope of tag disambiguation. Hence, an automatic approach to select the correct concept from knowledge data stores is more desirable for tag disambiguation. This automatic procedure is not a straight forward task since intelligent systems must be developed with great care in order to infer the correct concept for the tagging activity. Moreover, retrieving concepts from knowledge data stores, both in automatic or non-automatic approaches to match a concept with a tag, is neither an easy task due to word synonyms, word hypernym, different variations of spelling or words that are meaningless. Therefore, a proposed automatic approach such as FLOR [15], takes these issues into consideration when adding concepts to tags. Such systems propose an approach that consists of various steps. The steps employed by FLOR are summarised as follows:

- The first step is lexical processing whereby on the basis of heuristics the input tag set is filtered and meaningless tags are excluded from the set.
- The second step is sense definition and semantic expansion whereby a WordNet sense is assigned to each tag based on its context with relation to other tags. Moreover, all relevant synonyms and hypernyms are extracted for a richer representation of the tag.

- The final step is Semantic Enrichment whereby each tag is associated with the appropriate Semantic Web Entity (SWE), which is an ontological entity defined in an online available ontology.

Most of these approaches use clustering algorithms in order to create clusters of related tags. These clusters are based on the co-occurrence of tags. Although clustering algorithms are efficient algorithms for organising such co-occurrence relationships amongst tags, the algorithms employed do not unfold new relationships amongst tags. In some instances, even though tags co-occur with each other, these tags might not be related and can produce inaccurate searches. Therefore, it is worth using an algorithm that unfolds new relationships besides the co-occurrence of tags that can result in more meaningful relationships amongst tags. Apart from unfolding new relationships, the algorithm must compute the relationships in an efficient manner by not being time consuming. Hence, an algorithm must be selected that produces good relationships amongst tags in the least possible time.

#### 4. FAST UNFOLDING OF COMMUNITIES IN LARGE NETWORKS

An approach to relate tags together is by using graph clustering techniques. An algorithm that produces prompt results is the fast unfolding of communities in large networks defined in [19]. In fact, this work shows that this process outperformed other methods and the algorithm managed to identify communities in a 118 million nodes network in 152 minutes. This algorithm shows that it finds high modularity partitions of large networks and unfolds a complete hierarchical community structure for the network. This algorithm proposes a rapid and easy method to extract community structure from large networks based on modularity optimisation. The modularity calculates the quality of a partition of a network. This quality value “is a scalar value between -1 and 1 that measures the density of links inside communities as compared to links between communities”.

This algorithm is recursive and each iteration is called a “pass”. The height of the community is determined by the number of passes executed. Each pass consists of two phases. The first phase consists of assigning a community to each node of the network and therefore each node in the network represents a community. Once the nodes are identified as communities, for each node, the gain in modularity is evaluated if the node is placed within the neighbour’s community. If the gain in modularity is positive, then the node is placed in the neighbour’s community. However, if the gain in modularity is negative, then the node is not moved. This process is repeated for all nodes until no further improvements can be achieved. When no further movements can be accomplished, then the first phase is complete. The second phase of the algorithm consists of building a new network composed of those nodes that are now communities found in the first phase. This is achieved by adding up the weight of the edges between nodes in two respective communities that are now new nodes in the new network. Once the second phase is over, this denotes a pass. The algorithm then applies the first phase on the resulting new network and to iterate the process. The whole process is iterated until there are no more improvements and the maximum modularity is achieved.

## 5. STATE OF THE ART APPLICATION

In the above sections it was noted that tags are a rich classification scheme for organising content. It was also noted that tags can be very powerful when searching for content since such tags are used in conjunction with other tags and thus relationship patterns amongst tags can be inferred. Moreover, since multiple resources are tagged with similar tags, tags can be used to relate resources together and hence creating a network of linked resources. It has also been noted that to harness the advantages provided by tags, tags have to be represented semantically in order for machines to manipulate such tags. Since widely used semantic representations of tags are already employed, such as the ones mentioned in the previous sections, it is worth using one of these representations. In this paper it was also noted that by finding new relationships of tags apart from their co-occurrence with other tags can contribute to better search results since tags can relate better to other tags than the ones they have been tagged with. The algorithm fast unfolding of communities in large networks, as explained above, has shown that it can produce good communities of networks in the least time compared to other methods. In order to use these new relationships for searching content and for inferring other knowledge, these relationships must be semantically enriched. Therefore, this research proposes a state of the art application that by using Semantic Web technologies transforms the hierarchical community structure produced by the fast unfolding of communities in large networks algorithm into a semantic representation. The application also provides various functions to persistently store these semantic representations on backup storage. Moreover, it employs Semantic Web query methodologies to retrieve the relationships from such hierarchical structure.

### 5.1 Dataset

Prior to this research, co-occurrence of tags were extracted from the Bibsonomy website. These tags were then used to compute relationships in the form of communities by using the fast unfolding of communities in large networks algorithm. This algorithm produced several edge list files that each file represents a pass from the algorithm. These edge list files consist of communities related to other communities together with the weight of the relationship. These edge list files were used as the sample dataset on which the application was developed. The dataset consists of four edge list files:

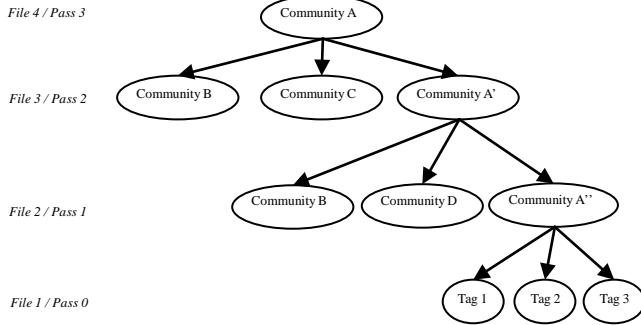
- The first file represents the co-occurrence of tags extracted from Bibsonomy. These tags are found at the lower level of the hierarchical community structure.
- The other three files represent the respective passes performed from the unfolding of communities algorithm that denote the communities relating to other communities.

Table 1 depicts the number of nodes in each of the edge list files and also how each line in the edge list files is formatted. In the first file, each line denotes a relation in the format  $\langle tag_i, tag_j, weight \rangle$  and in the other three files the relation is in the format  $\langle community_i, community_j, weight \rangle$ . These relations state that the two nodes, either tag nodes or community nodes, relate to each other and the relationship value

is denoted by the weight value. Figure 1 shows an illustration of a tree that depicts the concept of the hierarchical community structure that can be formed from the edge list files.

**Table 1. The number of nodes and the format of each line in each edge list file**

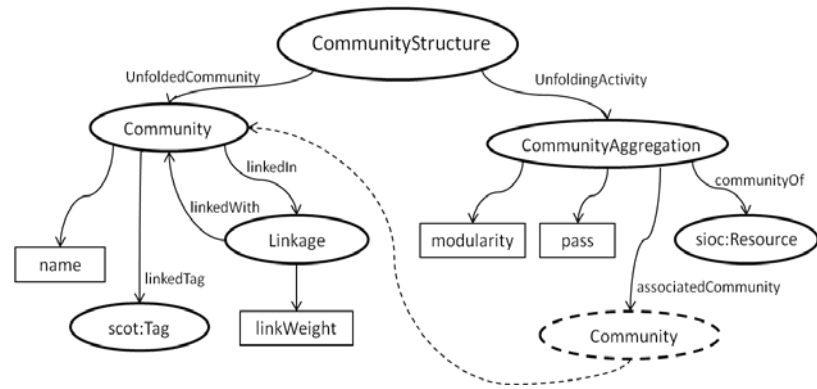
Edge List File	Pass	No. of Nodes	No. of Edges	Data Format
1 <sup>st</sup>	-	13,126	264,718	<tag <sub>i</sub> , tag <sub>j</sub> , weight>
2 <sup>nd</sup>	1 <sup>st</sup>	529	6,337	<community <sub>i</sub> , community <sub>j</sub> , weight>
3 <sup>rd</sup>	2 <sup>nd</sup>	65	374	
4 <sup>th</sup>	3 <sup>rd</sup>	50	207	



**Figure 1. Edge list files hierarchical structure**

## 5.2 Community Structure Ontology

Once the hierarchical structure of the edge list files can be visualised, an ontology for this structure can be designed. The hierarchical structure contains communities that link to other communities or to other tags. As regards to semantically describing the relationships amongst tags, the tag ontologies explained earlier are already established and widely used. Therefore, it is worth using one of these ontologies to semantically describe the relationships between tags. Since in the edge file a relation exists amongst the tags, it is worth using the SCOT ontology. This is because the SCOT ontology contains classes and properties that can describe tags and the relationship between co-occurring of tags. As regards to semantically describing the communities that relate to other communities or tags, a new ontology has to be designed and aligned with the SCOT ontology to allow describing the relationship between a community and a tag. This means that the new ontology will contain a property that will describe the relationship between a community and the class `scot:tag`, and the class `scot:tag` is used to describe the tag. Figure 2 illustrates the community structure ontology that was constructed using OWL. Table 2, table 3 and table 4 explain the composition of this ontology by explaining the classes, object properties and data properties respectively. This ontology was designed using Protege<sup>8</sup>, a free open source ontology editor.



**Figure 2. Community Structure Ontology**

**Table 2. Community Structure Ontology Classes**

Class	Explanation
Community Structure	An object that describes the activity of unfolding community structures.
Community	An object that describes a community which can be related to other communities and can be related to tags.
Linkage	An object that describes the relationship between two communities.
Community Aggregation	An object that describes the aggregation of communities.

**Table 3. Community Structure Object Properties**

Class	Explanation
Unfolded Community	The relationship between the community structure and a community.
unfolding Activity	The relationship between the community structure and an aggregated community.
linkedIn	The relationship between the object community and the object linked communities.
linkedWith	The relationship between the object linked communities and the object community.
linkedTag	The link between a tag ( <code>scot:Tag</code> ) and a community.
communityOf	The relationship between a resource ( <code>sioc:item</code> ) and a community.
Associated Community	Indicates the aggregated communities.

**Table 4. Community Structure Data Properties**

Class	Explanation
communityName	The name of a community.
linkWeight	The weighted value of the link between two communities.
modularity	The modularity value of an aggregated community.
pass	The total number of passes of an aggregated community.

<sup>8</sup> Protege: <http://protege.stanford.edu/>

### 5.3 Creating Storing and Querying RDF/XML Statements

With the community structure ontology, the edge list files can now be parsed and represented in RDF statements serialised in RDF/XML adhering to the ontology. The state of the art application contains the functionality to parse the edge list, create RDF statements, store these statements in backup storage for later retrieval and query RDF statements.

#### 5.3.1 Parsing Edge Lists

Prior to creating the RDF/XML statements, the application first merges all the edge lists into one structure and orders the structure to have the communities structured as depicted in figure 1. Once the edge list is ordered, then the RDF/XML statements can be created.

#### 5.3.2 Creating RDF/XML Statements

In order to create the RDF/XML statements, the application makes use of Jena<sup>9</sup>. Since the application is developed using the Java programming language, Jena provides a java framework for building Semantic Web applications. The Jena framework provides an RDF API for reading and writing RDF statements serialised in various formats including RDF/XML serialisation. Apart from the RDF API, Jena provides an OWL API that provides a programming interface for ontology development. Moreover, Jena provides in-memory and persistent storage capabilities and also a SPARQL query engine. Therefore, to create RDF statements for the edge list, by using Jena's APIs, first the community structure is loaded and the ontology classes and properties are represented with Java classes. This allows the creation of the RDF statements to adhere to the community structure ontology.

#### 5.3.3 Storing RDF/XML Statements

As mentioned in the previous section, Jena provides various methods for storing RDF statements. Jena provides in-memory storage and also backup storage for RDF models. The backup storage that can be used through the state of the art application are: text files with the extension .rdf, MySQL<sup>10</sup> relational database using Jena's RDB system to access relational databases, and Jena's TDB system which is a native persistent engine using custom indexing and storage. The application provides the user to setup the type of persistent storage preferences and according to what the user selects, the application uses the required Jena system to write the RDF statements. The text file method provides two storage methods, either as a whole document that contains the whole hierarchical structure in RDF/XML statements or else in split text files that each file contains the structure for one community. The split text file also contains indexes for fast retrieval of community files.

#### 5.3.4 Querying RDF/XML Statements

Querying RDF statements depends on the type of backup storage used. For text files with the extension .rdf, another SPARQL

engine called Corese is used. This is because Corese<sup>11</sup> provides additional enhancements over the standard SPARQL engine such as approximated searches and select expressions. When the RDB persistent storage system is selected, Jena's SPARQL engine is used since Jena provides direct access to relational databases for SPARQL queries. For the TDB system, since it is a custom persistent storage provided by Jena, in order to access such structure, Jena's SPARQL engine is used since it allows direct access to TDB persistent storage. The SPARQL queries are built in the state of the art application and passed to the respective SPARQL engine. The querying consists of the user giving a search term and the application retrieves the related community names and tags that are linked to that term. Therefore, the query result consists of a list of related community names and related tags for that search term.

#### 5.3.5 Results

The application provides a convenient solution to semantically represent tags and their relationships unfolded using the fast unfolding of communities in large networks algorithm. As noted in the previous sections, the application provides the creation of RDF/XML statements adhering to the community structure ontology. Table 5 shows the duration for each storage method used by the application to create 376,969 RDF/XML statements. From this table it can be noted that the RDB method took the longest to complete. It is worth mentioning that Jena provides another system called SDB that it is an enhancement over the RDB method. It could be that if the SDB system was used, the duration would be much less than the RDB system for writing statements to relational databases. Moreover, from these results, the writing to text files yields the lowest results.

**Table 5. The duration to write RDF/XML statements**

Storage Method	Duration
RDF Documents (Whole Documents)	1.9 minutes
RDF Documents (Split Documents)	2.2 minutes
RDF Persistent Storage (RDB Method)	35.2 minutes
RDF Persistent Storage (TDB Method)	3.2 minutes

Table 6 shows the duration for each storage method used by the application to query a community that is linked to 57 other communities and linked to 15 tags. From these results, contrary to the results obtained when writing RDF/XML statements, the RDB and TDB systems yield the lowest results. When querying the RDF whole document, the application could not retrieve the results since the maximum amount of memory was reached. As can be concluded from these results, the optimal solution for creating and querying the RDF/XML statements on this dataset is by using the TDB method.

**Table 6. The duration to query RDF/XML statements**

Storage Method	Duration
RDF Documents (Whole Documents)	Out of Memory
RDF Documents (Split Documents)	43.3 minutes

<sup>9</sup> Jena – A Semantic Web Framework for Java: <http://openjena.org/>

<sup>10</sup> MySQL: <http://www.mysql.com/>

<sup>11</sup> Corese: <http://www-sop.inria.fr/edelweiss/software/corese/>

RDF Persistent Storage (RDB Method)	3 seconds
RDF Persistent Storage (TDB Method)	1 second

## 6. FUTURE ENHANCEMENTS

The future of the Semantic Web lies in linked data whereby different data sources can be linked together. Many Web applications provide APIs to link to their data sources but this involves many customisations for every specific API. The Linking Open Data (LOD) community project<sup>12</sup> is an initiative that is encouraging online websites to make available their data sources using Web of data standards such as RDF. Moreover, vocabularies help to define the RDF documents that would ease data sources to interoperate with each other. With respect to the Social Web sites interoperating together; the semantic web provides frameworks such as Semantically Interlinked Online Communities (SIOC)<sup>13</sup>, Friend-Of-A-Friend (FOAF)<sup>14</sup> and Simple Knowledge Organisation Systems (SKOS)<sup>15</sup>. SIOC provides a generic ontology that defines main concepts and properties required for representing data about the structure and contents of Social Web sites in RDF. Moreover, SIOC provides a means for Social Web sites to find related content information and to help link with other content items and community objects. FOAF provides a generic ontology that describes person-related data. SKOS provides a vocabulary to define the basic structure and content of semi-formal knowledge organisations such as thesauri, classification schemes, subject heading lists, taxonomies, folksonomies and other similar controlled vocabularies. Since it is designed on RDF, SKOS allows these semi-structured concepts to be published on the Web, linked to data available on the Web and also incorporated with other concept schemes. SIOC, FOAF and SKOS can therefore be implemented into the proposed state of the art application to define better the tagging activity to be interlinked with other Social Web sites. For instance by using the SIOC ontology to represent the tags as some sort of post in an online community platform, this would enable the tags to be sharable with other content. In addition to this, the person who tagged the tags would also be implemented in the state of the art application and defined using the FOAF ontology.

## 7. CONCLUSION

This paper portrayed the importance of publishing data in the semantic format RDF for the purpose of adding metadata to data that can be processed by machines to infer further knowledge. These RDF documents, defined by ontologies which in themselves are common agreed upon concepts, make the RDF documents more accessible on the Web that can be linked to other data sources represented in semantic formats. The importance of tags has also been illustrated in this paper as a means to search and retrieve content more intelligently. It is

<sup>12</sup>The Linking Open Data Project:

<http://esw.w3.org/topic/SweoIG/TaskForces/CommunityProjects/LinkingOpenData>

<sup>13</sup>SIOC: <http://sioc-project.org/>

<sup>14</sup>FOAF: <http://www.foaf-project.org/>

<sup>15</sup>SKOS: <http://www.w3.org/2004/02/skos/>

therefore extremely significant to publish tags in a semantic format that would enhance searching capabilities. This paper therefore proposed an ontology and a state of the art application that would transform tags produced from the fast unfolding of communities in large networks algorithm into a semantic format that can be queried upon to produce more accurate results.

## 8. REFERENCES

- [1] Fielding R., Irvine UC, Gettys J., Mogul J., Frystyk H., Masinter L., Leach P., Berners-Lee T. 1999. RFC 2616 – Hypertext Transfer Protocol – HTTP/1.1. [Online]. <http://tools.ietf.org/html/rfc2616>
- [2] Vander Wal T. 2007. Folksonomy Coinage and Definition. [Online]. <http://vanderwal.net/folksonomy.html>
- [3] Murugesan S., 2007. Understanding Web 2.0. In: IT Professional, IEEE, 9(4): 34-41.
- [4] Golder S., Huberman B.A.: The Structure of Collaborative Tagging Systems. HP Labs technical report. [Online] <http://www.hpl.hp.com/research/scl/papers/tags/>
- [5] Alpert J., Hajaj N. 2008. We knew the web was big. The Official Google Blog. [Online] <http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html>
- [6] Newman R. 2005. Tag Ontology. [Online] <http://www.holygoat.co.uk/projects/tags/>
- [7] Gruber T. 2007. Ontology of Folksonomy: A Mash-up of Apples and Oranges.
- [8] MOAT: Meaning Of A Tag. [Online] <http://moat-project.org/>
- [9] SCOT: Let's Share Tags. [Online] <http://scot-project.org/>
- [10] CommonTag. [Online] <http://www.commonitag.org/Home>
- [11] Spyns P., de Moor A., Vandenbussche J., Meersman R. 2006. From Folkologies to Ontologies: How the Twain Meet. In: Meersman R., Tari Z. Et al. (Eds.), OTM2006, LNCS 4275, pp.738-755.
- [12] Specia L., Motta E. 2007. Integrating Folksonomies with the Semantic Web. In: Franconi E., Kifer M., May W., (Eds.), ESWC 2007, LNCS 4519, pp. 624-639.
- [13] Sabou M., d'Aquin M., Motta E. 2008. Exploring the Semantic Web as Background Knowledge for Ontology Matching. In: Spaccapetra S. et al. (Eds.), Journal on Data Semantics XI, LNCS 5383, pp. 156-190.
- [14] Angeletou S., Sabou M., Specia L., Motta E. 2007. Bridging the gap between folksonomies and the semantic web: An experience report. In: Proc. Of the 4<sup>th</sup> ESWC, pp 624-639.
- [15] Angeletou S., Sabou M., Motta E. 2008. Semantically enriching folksonomies with FLOR. In: 5<sup>th</sup> ESWC.
- [16] Angeletou S., Sabou M., Motta E., Folksonomy Enrichment and Search. In: Aroyo L. et al. (Eds.), ESWC 2009, LNCS 5554, pp.801-805.
- [17] Van Damme C., Hepp M., Siorpaes K. 2006. FolkOntology: An Integrated Approach for Turning Folksonomies into Ontologies.
- [18] Lin H., Joseph D., Zhou Y. 2009. An Integrated Approach to Extracting Ontological Structures from Folksonomies. In: Aroyo L. et al. (Eds.), ESWC 2009, LNCS 5554, pp.654-668.
- [19] Blondel V.D., Guillaume J.L., Lambiotte R., Lefebvre E. 2008. Fast unfolding of communities in large networks.