Create Semantic Annotation for Web Contents with Triple Tagging

Jie Yang
The University of Tokyo
yangj@mi.ci.i.u-tokyo.ac.jp

Yutaka Matsuo
The University of Tokyo
matsuo@biz-model.t.u-tokyo.ac.jp

Mitsuru Ishizuka
The University of Tokyo
ishizuka@i.u-tokyo.ac.jp

Abstract

An important issue of the Semantic Web is to create semantic metadata that describes the contents of web resources. Creating semantic annotations according to the predefined vocabulary (i.e., ontology) is a tedious process. Recently, social tagging, an activity that users classify web contents by self-chosen keywords (which is called tags) has become immensely popular. It is regarded as a valuable source of semantic annotations by the Semantic Web society. Inheriting the idea of social tagging, this paper proposes triple tagging, a social annotation diagram, where users annotate web contents by (subject, predict, object) triples. The triple structure has richer expressivity than current tagging scheme and enhanced flexibility than annotation methods that requires predefined vocabulary. We provide a formal description of triple tagging scheme and develop a triple tagging ontology with Semantic Web technologies. We present by examples how the triple tagging ontology is used in the social annotation environment. Furthermore, we build a triple tagging system, Tripe-Note. Users of the system can triple annotate web pages collaboratively and browse (personal or social) triple tags graphically. Triple tag browsing provides the immediate feedback which is important to collaborative systems. Filter and query functions are provided to facilitate the triple browsing. We believe that triple tagging provides a valuable and practical link between social tagging activity and the Semantic Web.

1 INTRODUCTION

The Semantic Web [5] is an extension of the World Wide Web where information is understandable to computers. A key issue for the Semantic Web is to enrich the human-understandable web contents with machine-understandable metadata. The process is generally referred to as the semantic annotation task, and semantic metadata in this case is referred to as semantic annotations.

From the Semantic Web point of view, metadata being ”machine-understandable” means it describes information of web contents using vocabularies defined in ontologies. Ontology in information science, deeply rooted in logic, is an explicit specification of conceptualization. Applications, by committing to the same ontologies, can understand the metadata and perform formal operations, such as inference, over it. Consequently a semantic annotation task generally covers three aspects: ontologies that provide formal semantics, metadata that instantiates ontologies, and potential semantic applications that consume metadata. We call annotation methods developed under this view the ontology priority methods.

In the ontology priority methods, first ontologies are defined by domain experts and ontology engineers; then metadata for web contents is created using vocabularies from the predefined ontologies. Both manual [12, 10] and (semi-) automatic [14, 7, 13] methods are explored [20, 23]. Ontology priority methods work well within the anticipated domain, but are argued to be insufficient when people try to describe information across the Web [17]. On the one hand, ontologies adopt a model-theoretic definition of meaning [8], which does not provide the rich presentation ability compared to, for example, natural language contents; and suffers from an inability to represent exceptions to logic rules and the contexts in which they’re valid [17, 11]. On the other hand, it’s a time-consuming task for application builders and (especially) web users to investigate and find out appropriate ontologies for annotation. As a result, so far we still observe a lack of both semantic metadata and therefore semantic services.

In response to the challenge, the lower-case semantic web is proposed. In the vision of the lower-case semantic web, three aspects of semantic annotation task are: human-meaningful metadata which is created and shared by users, applications that enable users to consume the metadata, and emergent semantics which is created ad-hoc and evolved over time. We call annotation methods following the lower-case semantic web vision the metadata priority methods. Recently popularized social tagging systems (e.g., del.icio.us, flickr

and CiteULike) are viewed as metadata priority practices, in which users attach metadata in the form of keywords (called tags) to classify web resources (such as pictures, web pages, blogs and video clips) and share them with other users. Many services are implemented by tagging systems. Their features are summarized into immediate self feedback (retrieval of one’s historical data) and social feedback (serendipity and browsing) \(^2\). To derive the emergent semantics for tags, a tripartite model consists of web resources, users and tags is proposed [18]. Many metadata priority annotation methods have been proposed (for example, extended tagging models [16, 22] and semantic wiki systems [15, 2]), as well as methods that derive emergent [26, 4, 25, 1].

This paper proposes a metadata priority annotation diagram which allows users to collaboratively annotate web contents with (subject, predict, object) triples. We call the annotating action triple tagging because, like the tagging activity, users are free to choose words for triple elements (i.e. subject, predict or object). Accordingly, triple annotations are called as triple tags. A triple is a human-meaningful statement about the web content, which means that there is a relationship presented by the predicate between the object and the subject. For example, the triple tag (Smashing Pumpkins, origin, Chicago) is interpreted as “Smashing Pumpkins is originated from Chicago”. Users who triple-tag web contents are supposed to accept the specified syntax and semantics. The triple is a building block of knowledge representation techniques such as semantic net, frame and the Semantic Web; hence triple tags can serve as raw data for knowledge base construction.

This paper presents a triple scheme, and we develop a triple tagging ontology with Semantic Web technologies. Therefore existing Semantic Web techniques such as SPARQL\(^3\) can be used seamlessly. Base on the triple tagging ontology, applications can collect, merge, query and disambiguate the meaning of triple tags. Thus services providing self and social feedback can be implemented. Existing methods such as the tripartite model [18] can be exploited. A collaborative annotation system Triple-Note is implemented. Users can triple-tag web contents and share the tags; triple tags can be browsed graphically; filter and query mechanisms are provided for efficient browsing.

2 RELATED WORK

There are notably many approaches that explore social strength to generate machine readable metadata of web contents. Existing tagging systems provide metadata in the form of single tags, where a tag provides a faceted classification about the topic of the web resource. A web source can be tagged by several words (tags), however the tags do not necessarily have semantic relations with each other. Compare to the triple tag scheme proposed in this paper, three tag elements of a triple tags are grouped semantically, and they act as a statement, instead of a topic class of the web content.

The trend of employing rich tag structure like microformat tagging has been discussed [3]. Microformat, such as GeoTag \(^4\), provides a simple structure and small vocabulary, which can be thought of as a lightweight ontology. Essentially, microformat based tagging follows the ontology priority annotation method. Also, there are on-going studies that propose enriched tag models [16, 22]. SemKey [16] tends to solve the semantic clarity problem of social tags and proposes a solution that connects tags to concepts in the WordNet. Extreme Tagging [22] extends the tag structure so that keyword tags can be tagged further. Although their tagging schema and ours share some similarities, our work focuses on providing an expressive annotation model, instead of classifying the tag semantics.

Other social annotation practices include Semantic Wiki systems [15, 2]. However, they introduce the concept of ontology to some extent and the semantic metadata are embodies into the page content. We formalize the concept of triple tagging and model it with RDF. Related efforts include [9, 19] and SCOT \(^5\). The fundamental difference of our work from theirs is that we are modeling the new tagging diagram - the triple tagging; whereas, they model the single tagging diagram.

3 TRIPLE TAGGING MODEL

This section first formalizes the concept of triple tagging and introduces the triple tagging ontology. Then we show by examples how triple tags are presented and merged using the ontology.

3.1 Triple Tagging Ontology

A triple tag is a RDF-like statement. Let \(W = \{w_1, w_2, w_3, \ldots \} \) be a set of terms (words or combination of words) , the triple tag is defined as:

**Definition 3.1 (Triple tag)** Syntaxically a triple tag \(t = (w, e, w')\), where \(w, e, w' \in W\). The semantics of the triple is: some relationship, indicated by term \(e\), holds between term denoted by \(w\) and \(w'\).

In this paper we also call \(w\) the subject, \(e\) the predicate and \(w'\) the object. Users who triple tag web contents are supposed to accept the specified syntax and semantics.

\(^2\)http://www.iskoi.org/doc/folksonomies.htm
\(^3\)http://www.w3.org/TR/rdf-sparql-query/
\(^4\)http://en.wikipedia.org/wiki/Geotagging
\(^5\)http://scot-project.org/about/
Definition 3.2 (Tag graph) The tag graph $G = \{t_1, t_2, t_3, \ldots\}$ is defined as the set of triple tags. For $\forall t = (w, e, w')$, where $t \in G$, $w$ and $w'$ are defined as the nodes of $G$, $\langle w, e, w' \rangle$ is the directed edge connecting node $w$ and $w'$, and $e$ is the label of the edge.

The action that users annotate web resources with triple tags is called triple tagging. Being a web content annotation diagram, we argue that users should be able to annotate inner pieces of web content, such as a sentence in a web page, a part of a picture. As is presented by definition 3.3 we refer to the web resource by its URI and record the inner piece in the content.

Definition 3.3 (Triple tagging) Triple tagging is modelled as a 4-tuple $a = < g, u, r, c >$, $a \in A \subseteq G \times U \times R \times C$, $g \subseteq G$ is a tag graph. Set $U$ denotes the set of the users, $R$ the set of URI, and $C$ the set of contents residing in resources identified by URIs.

Applications are always making some assumptions on metadata semantics in order to process them “wisely”, either explicitly or implicitly. Existing tagging systems assume that tags with the same string name have the same meaning. Moreover, to provide advanced services like inference, it is necessary to formalize tag elements as formal concepts. To assert the meaning of a tag element explicitly we introduce the concept of semantic identification.

Definition 3.4 (Semantic identification) The semantics identification function $\text{sem} : W \times U \rightarrow R$ means that user $u \in U$ asserts that the tag element $w \in W$ has the meaning presented by $r \in R$.

The resource $r$ stands for an arbitrary resource. It can be either a formal concept or informal object. The function is featured by our conception that the tag element $w$ is just a symbol whose meaning varies according to different user $u$.

We model the above concepts into a triple tagging ontology. Figure 1 depicts the ontology in the form of RDF graph. The triple tag is constructed by the TagElement class. Given a triple tag $\langle w, e, w' \rangle$, $w$ and $w'$ are instances of class TagElement, which are connected by the property TagElement $e$, as is shown in box A of figure 1. Note that the TagElement acts as both the class and property thus our ontology falls in the reign of OWL full. TagElements constitute a TagGraph. The triple tagging action is modeled around the Context class (shown in box B). A context consists a user (FOAF:Agent) who makes the semantic assertion can be described. Examples on the usage of the triple tagging ontology are given in the next section.

3.2 Example of Triple Tagging Ontology

3.2.1 Triple Tagging

The upper part of figure 5 is a partial page snapshot from wikipedie. It presents information of a rock band. Suppose user $u$ triple annotates the web page by follow triples:

(\text{Smashing Pumpkins}, \text{isa}, \text{rock Band})(\text{Smashing Pumpkins}, \text{start from}, 1988)

Figure 2 shows how the triple tagging information is instantiated with the triple tagging ontology. In figure 2, first six RDF resources are created. They are instances of the TagElement class. Each element is assigned a URI so that it is globally unique and globally accessible. For example, the first tag element is assigned the URI of http://www.miv.t.u-tokyo.ac.jp/tripletag#TagElement_1. The URI of a tag element can be generated, for example, by combining the identification of the user (e.g. encrypted email address) and the time that the tagging action takes place. Tag elements also have labels recording their string names (e.g., \text{Smashing Pumpkins}). Six tag elements form two triple tags, as is shown in the middle part of figure 2. The graph sampleGraph1, containing the two triples appears in the sampleContent_1. The content records the scene of the triple tagging action, including information about time and tagger. Note that although some tag elements have the same string name (e.g. "..."), they are treated uniquely.
Currently one mechanism is shown to work effective in cur-

3.2.2 Collaborative Tagging

Suppose another user user2 triple-tags the same page by

\begin{quote}
(Smashing Pumpkins, genre, Punk rock)(Punk rock, is a, rock)(Smashing Pumpkins, has member, Billy Corgan)(Billy Corgan, plays, Guitar).
\end{quote}

Figure 3 presents partially the triple tagging information of user2, which is similar to the content shown in figure 2. Thanks to the data integration capability of RDF\(^7\) and existing Semantic Web technologies such as RDF query languages (SPARQL), we can merge tag graphs of different users easily. More specifically, since sampleGraph_1 in figure 2 and sampleGraph_2 in figure 3 annotate the same web page (the taggedObj properties have the same value), we retrieve triples of the two graphs (which contain TagElement_1 to TagElement_15) and construct a new graph for the same page. In the collaborative annotating diagram, a user can just annotate a part of a web resource that he is interested in. Metadata from different users will complement each other. For example, user2 gives member information of the band, which does not appear in user1's triples.

3.2.3 Semantic Identification and Machine Processability

Figure 4 shows how to assert the semantics of tag elements. The technique we use here is called RDF reification. Three statements are shown in figure 4. The first statement claims that tripleTagAgent_1 thinks resources TagElement_1 and TagElement_4 have the same meaning. Referring to the previous example, they have the same surface name, Smashing Pumpkins. The second statement claims that user1 says TagElement_2 (isa) and TagElement_11 (isa) means the same thing. And by the third statement, agent_2 asserts that TagElement_2 semantically equals to owl:sameAs. Figure 4 shows that the same tag element TagElement_2 is assigned different semantics by different users.

How to model, discover and exploit the emergent semantics from social annotation is still a heated ongoing study. Currently one mechanism is shown to work effective in cur-

\footnote{http://www.w3.org/TR/rdf-dawg-uc/\#d4.2}
client side user interface is implemented by a Mozilla Firefox, including, users can annotate arbitrary web page; triple tagging scheme called Triple-Note. Triple-Note takes the client-server architecture. Its details are introduce in the following subsections.

4 IMPLEMENTATION

We implemented an experiment annotation system of the triple tagging scheme called Triple-Note. Triple-Note takes the client-server architecture. Its details are introduced in the following subsections.

4.1 User Interface

We design the user interface with following principles. Firstly, as a social annotation system, tagging action should be kept simple and easy. Several criterion should be satisfied, including, users can annotate arbitrary web page; triple tagging a web page does not force users to leave the current page; tagging action should be no more than inputing a few words. Take these requirements into consideration, the client side user interface is implemented by a Mozilla Firefox extension. Secondly, users should be able to consume triple annotations they created, in the sense of both self and social feedback. To this purpose, we implemented a triple browser which presents triple tags graphically. Users can either browse triple graph, or visit the web resources annotated by triples. Filter and query mechanisms are implemented to facilitating the tag graph browsing process.

The user interface of Triple-Note is shown in figure 5. By default it is hidden. When the label Triple-Note on the status bar is clicked, the triple-note dialog pops up. There are two tab-panels shown in the dialog, namely, TiTagging (stands for triple tagging) and TiBrowsing (stands for triple browsing). Figure 5 shows the TiTagging tab-panel, and figure 6 shows the TiBrowsing tab-panel.

4.1.1 Triple tagging web contents

Users annotate web contents using the TiTagging tab-panel. To triple tag a web page, the user clicks the load button in the leftmost panel, then the URI and title of page will be loaded. The user can input self-defined triple tags to the text box of the rightmost panel. In figure 5 there are four user-added triple tags which annotate the current web page. By clicking the Save to Server! button, the tagging information is sent to the server and saved. The Triple-Note server saves data in the form as shown in figure 2.

4.1.2 Triple recommendation

Optionally, Users can also annotate the inner sentences of a web content. The user can select sentences in a web page then by pressing the load button, the selected sentences are also loaded to the leftmost panel along with the URI and title information. In figure 5, the sentence “The Smashing Pumpkins are an American alternative rock band that formed in Chicago, Illinois in 1988” is selected and loaded.

At this time, the user can ask for the recommended triples from Triple-Note server. Clicking the Fetch! button, the selected sentence is sent to the Triple-Note server. At the server side, sentence are passed to triple extraction module [27], which extracts triple tags automatically. Extracted triples are sent back to the client and listed in the middle panel. If a triple is thought be to useful, by a click it will be copied to the rightmost panel. Two triples are shown in the second panel of figure 5. The first one is copied to the rightmost panel.

4.1.3 Triple graph browsing

We implemented a triple tag browser based on SVG\(^8\) technique. Triple tag browser organizes and presents triple tags graphically. The subject and object elements act as nodes of the graph. The predicate element acts as the edge. Triple tag browser judges semantics of tag elements by their string

\(^8\)http://www.w3.org/Graphics/SVG/
names, as is discussed in section 3.2.3: If nodes from different triples have same string names, they are merged and shown as one. There are two areas show in TiBrowsing tab-panel. The tag graph is shown in the right (main) area. Graph elements (nodes and edges) are mouse selectable. Clicking a graph element, a list of URIs tagged by the element will shown in the left area, as well as the creators (taggers) of the element. Figure 6 exhibits the merged tag graph of the current web page. Three annotators are shown in the right panel. The graph on the one hand shows the collective knowledge constructed by triple tags, on the other hand provides a new way of browsing web information.

Tag graph can be filtered from two aspects. Firstly, by checking the “Triples of current page” checkbox, only triple tags of the current page will be shown. Secondly, by checking the user name, the selected user’s triple tags will be retrieved and presented graphically. In addition to filter function, query function is also provided, as is introduced below.

4.1.4 Triple querying

To make triple tag browsing more efficient and flexible, the triple query function is provided. By input triple queries, user can browse a particular part of tag graph. A triple query in fact is a sequence of triple tags. Triple tags used
for query are especial in that the character * is reserved as the wildcard character. For example, the query (*, isa, rock Band) means to find all rock bands. Query function is implemented by graph match. For example, the triple (Smashing Pumpkins, isa, rock Band) is a qualified candidate; the triple tag (Smashing Pumpkins, is_a, rock Band) is not. Because the predicates are different. A query might consist of more than one triples. In this case, those triples are connect by logic operator ADD. For example, given the query (*, isa, rock Band)(*, startfrom, 1988), Triple-Note system will start to find a rock band which is started in year of 1988. A convertor that converts triple queries into SPARQL queries is implemented and running on the server side. After find triples that meet the query, convert will return their 1-3 hop neighbors.

A query box is shown in the right area of the TiBrowsing tab-panel. By clicking the load button, user’s query is sent to the Triple-Note server and the result will be presented graphically in the browser. When the checkbox Triples of current page is checked, the query is only performed against the triples of the current page. Otherwise all triples in the system that meet the query will be returned. At present triple query only supports limited functions. More functions will be supported in the future.

It is worth pointing out how triple query works in the social annotation environment where no common vocabulary is provided. We first divide scenarios where triple query is used into self-triple query and social-triple query. In the former, the user is searching his own annotations, and he must know the vocabularies he used. In the latter there is no explicit defined vocabulary, however, we assume that users will use similar words for similar meaning. This trend is observed in existing tagging systems, as well as the primary experiment we conducted [27].

4.2 Architecture

Figure 7 gives the over all architecture of Triple-Note, which takes the typical client-server architecture. On the client side there are two modules, namely the firefox extension and the SVG based triple tag browser. On the server side, there are three application moduels, includ-
6 CONCLUSION AND FUTURE WORK

In this paper we proposed a triple tagging scheme to create semantic annotations for web contents. We defined the triple tagging ontology and presented by examples how the triple tagging ontology can be used in social annotation environment. A triple tagging system Triple-Note is implemented. The client side is implemented as a FireFox extension, users can triple tag web contents within the browser, explore triple tags graphically aided by filtering and querying mechanisms. Three modules are provided at the server side, namely the triple extraction module, the triple query conversion module and triple collection module. In the future, we will strengthen the triple extraction module, as well as the triple query converting module. If triple tagging achieves wild adoption, we will study the triple semantics and consensus knowledge constructed from triple tags.

References


