An Augmented Tagging Scheme with Triple Tagging and Collective Filtering

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

Yutaka Matsuo
National Institute of AIST
y.matsuo@aist.go.jp

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

Abstract

Collaborative tagging is increasingly drawing attentions. However the keyword based tagging scheme has its limitations and it can be observed that tagging society are seeking and using new tagging patterns. This paper proposes a subject-predicate-object *scheme for users to triple tag web resources. We first introduce the triple tag model and discuss its relations with existing tag schema and RDF model. Then a filter-based framework that supports the query of triple tags is proposed. The implementation and a case study in the comparison shopping domain are exhibited.*

1 Introduction

Collaborative tagging is the process by which users attach metadata in the form of tags to web resources and share them socially. Tags are keywords (or combination of keywords) chosen by users freely. Using tags, a user can not only organize personal data, but can also browse and search the shared information efficiently [4]. Already, many successful collaborative tagging systems are in use (e.g. Flickr, del.icio.us, and CiteULike)[1]. Collaborative tagging is considered as an important source that provides semantic metadata for web resources following the idea of emergent semantics [3]. Many studies have being carried out to clarifying the implicit semantics of tags [5, 7, 8] where tags are viewed as sets of keywords.

Recently, more and more evidences show that the keyword based tag scheme is not expressive enough and that users are seeking new ways to tag web resources with enriched semantics [2]. Users either use multiple tags as a group implicitly to convey complicated meaning (e.g. in flickr, the picture is tagged by *color* and *white* denoting that the *color* property of the picture is *white*.), or use structured tags (e.g. geotag[2]) that have enriched and predefined formats and semantics. The emergence of the rich-tagging patterns suggests that tagging community is expecting and ready for more expressive tagging scheme as long as instant benefits are provided. Some initial work providing rich structured semantic tagging has appeared [1, 6].

This paper proposes a unique *subject-predicate-object* scheme for users to triple tag web resources. The assertion of the schema says that the relation indicated by *predicate* holds between the *subject* and *object*. A query mechanism featured by collective filters is proposed as well. It allows users to find web resources by triple tags. Query mechanism is essential in the sense of completing the instant feedback loop which is the key to the wide adoption of collaborative tagging systems. The building element of the query framework is called a *triple filter* which can be implemented by users as required.

2 Triple Tagging

In this section we first represent what the triple tag is then introduce how it relates to the existing keyword based tag (which will be referred to as "tag" to be differentiated from the triple tag in the following discussion) and the RDF statement.

2.1 Triple Tagging Scheme

Let $W = \{null, w_1, w_2, w_3, \dots\}$ be a set of terms where $null$ is an empty term. The definitions of triple tagging scheme are given below.

Definition 1 (Triple tag) *Syntactically a triple tag* $t =< w, e, w' >$*, where* $w, e, w' \in W$.

Definition 2 (Tag graph) *The tag graph* $G = \{t_1, t_2, t_3, \dots\}$ *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,* $< w, e, w' >$ *is the directed edge connecting node* w *and* w' *and* e *is the label of the edge.*

The act that the user attaches a triple to a web resource is called triple tagging.

[1] www.flickr.com, del.icio.us, www.citeulike.org
[2] http://en.wikipedia.org/wiki/Geotagging

0-7695-3026-5/07 $25.00 © 2007 IEEE
DOI 10.1109/WI.2007.120

Definition 3 (Triple tagging) *Triple tagging is modelled as a 4-tuple* $a = < t, u, r, c >$, $a \in A \subseteq G \times U \times R \times C$. *Set* U *denotes the set of the users,* R *the set of information resources, and* C *the set of contents residing in the resources. The* tagged resources of t *is defined as a subset of* R *through function* $res(t) = \{r | \exists u \in U, < t, u, r, c > \in A\}$.

Different from most existing tagging systems where a tag is attached to the whole web resource, our triple tag scheme proposes to relate the tag not just to the web resource (r), but also optionally to its content(c), e.g. a sentence in the web page.

Definition 4 (Semantics assignment function) *Through the semantics assignment function* $sem : W \rightarrow R$*, the tag element* $w \in W$ *is assigned to possess the same semantics as* $r \in R$*, no matter whether* r *is formally defined or not.*

The resource r can be a concept, an instance of the concept, or another tag element. The dynamic assignment of the tag semantics is based on our conception that the tag is just a symbol whose meaning is vague and will change dynamically according to the user and circumstance.

The left hand side of figure 2 shows a web page about the mobile phone. The possible triple tags of the page are given in figure 1(a). Suppose the URL of the web page is *mobilephone.com* and the name of the user is *Tony*. The triple tagging information for t= <Screen, color, 65,636UBC> is shown in figure 1(b).

<Sony Ericsson, has, Camera>
<Sony Ericsson, -, Screen>
<Sony Ericsson, -, sound>
<Screen, color, 65,636UBC>
<sound, type, mp3/aac>

(a) Triple tags for the web page in figure 2

u = Tony
r =mobilephone.com
c ="65,536 color UBC (ultra bright color)"

(b) The triple-tagging information for tag t = <Screen, color, 65,636UBC>

Figure 1. Example: Triple-tagging information

2.2 Triple tag, tag and RDF statement

The triple tag scheme is backward compatible with the keyword based tag model. For example, suppose a web page is tagged by the keyword "phone". It can be rewritten as a <subject, predicate, object> triple of the form <*uri of web page, relevant-to, phone*>. The subject (*uri of web page*) and the predicate (*relevant-to*) are generated by the tagging system implicitly and users decide the object (*phone*) only. Some recent work [1, 6] allows users to assign predicate and object.

On the other hand, a triple tag syntactically equals to a RDF statement. They are different in that elements of a triple tag are the user-provided terms instead of URI references, blank nodes or literal. Thus triple tags can be converted to RDF statements seamlessly therefore we can realize various kinds of retrieval and inferences thanks to many studies of the Semantic Web data processing. For example, denote the prefix URI of an implementation tagging system as *system*, the triple tag <*screen, color,* $65,636$*UBC*> can be presented as a RDF statement of the form (we use "()" for RDF statement):

(system:screen, system:color, system:65,636 UBC).

Furthermore, with the semantics assignment function defined previously, the RDF triple above might be set as an alias for the following RDF statement whose semantics is defined in WordNet (*wn20instances* is the prefix of the WordNet namespace[3]).

(wn20instances:word-screen, wn20instances:word-color,
system:65,636 UBC),

To this end, the semantics of the original triple tag is understandable to other programs. Note that the triple model itself does not solve the semantic ambiguous problems that the keyword based tag model faces as well. There are various attempts towards clarifying tag semantics which we do not discuss in this paper.

3 Triple Querying

This section presents a personal filter based triple querying framework through which users can retrieve triple tagged web resources. The building component is the *triple filter*, which takes a query as input and returns a set of expected triples. It is pluggable in the framework and is created by users for their own purposes.

Definition 5 (Triple query) *A triple query* $q = \{t_1, t_2, t_3, \ldots\}$ *is defined as a sequence of triples, where* $t_i = < w, e, w' >$ *and* $w, e, w' \in W$.

Users are required to issue queries following the triple format as defined above. For example, in order to find web pages containing information suggesting that the price is lower than $500, the possible queries might be q = < price, =, $?><?, <,500>, or q =< price, lessThan, 500>.

Definition 6 (Triple filter) *A triple filter* f *consists of two functions, they are*

- ***Filtering function*** $filtering_f(G_{in}, q) = \{t | t \in G_{in}\}$ *takes a recognizable query* q *as input, finds the most suitable triples from the triple graph* G_{in}.

[3] http://www.w3.org/2006/03/wn/wn20/instances/

- ***commitment function*** $commitment_f(q) \in [0,1]$ *claims how well the filter f can accomplish the query task q. Filters with high commitments will be chosen for q and form the set of* ***Candidate filters*** F_q.

Definition 7 (Query result) *Denote* $T_q = \bigcup filtering_f(G_{in}, q)$*, where* $f \in F_q$*. The query result is defined as* $result(q) = \bigcup res(t)$*,* $t \in T_q$*.*

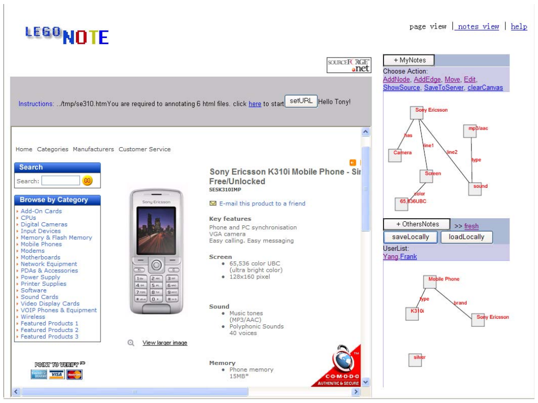
The result of a query is a set of web resources. Similarly, we define the **complex query** $Q = \{q_1, q_2, q_3, \dots\}$ that is a sequence of queries. And the **complex query result** $result(Q) = \bigcap result(q)$, $q \in Q$ which is the intersection of results from each sub query.

Given a query q, each filter reports its commitment ($commitment_f(q)$) and filters with highest commitments are selected. The triples returned by $filtering_f(G_{in}, q)$ functions of the selected filters are combined and $result(q)$ are returned to the user. In the process, the query, the filters and triple tags might probably come from different users. There are no predefined agreements on the semantics of the three parts. The framework is supposed to work under the premise that people will use similar triple for similar semantics. We are optimistic at this point, supported by experiences from existing tagging systems where the tag semantics from different users converges.

In section 5 we present examples of filter implementation and how they work in the query framework.

4 User Experience

This section introduces the user experience within our initial implementation *LegoNote*[9]. In LegoNote, users can create triple tags graphically and share them to the public, can register personal filters and search web resources through triple querying.



Instructions: ../tmp/se310.htmYou are required to annotating 6 html files. click here to start Hello Tony!

Sony Ericsson K310i Mobile Phone - Si Free/Unlocked
SESK310IMP

E-mail this product to a friend

Key features
Phone and PC synchronisation
VGA camera
Easy calling. Easy messaging

Screen
- 65,536 color UBC (ultra bright color)
- 128x160 pixel

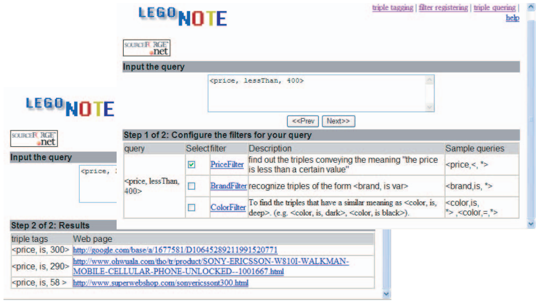
Sound
- Music tones (MP3/AAC)
- Polyphonic Sounds 40 voices

Memory
- Phone memory 15MB*

Figure 2. Screenshot of the triple tagging interface.

Figure 2 presents the user interface to tag a web page. The left-hand side of figure 2 is an embedded browser. Users load a page by either entering a URL or following the links in web pages. The triple tagging tool is floating on the right-hand side. In the upper area, users can create triple tags graphically. Basic graphic operations are provided showing them selves as links (*AddNode*, *addEdge*, *Move* and etc.). Users share triple tags by uploading them to the server. The bottom area exhibits tag graphs of the same page shared by other users. In figure 2 the web page about the cellular phone is loaded. Current user's tag graph is shown in the tagging tool. The other two users have shared their tags. One is shown graphically at the bottom area.

To register a filter three fields are required: the filter name, the URL to run the filter, and the description of its function. The implementation of a filter is loosely coupled with LegoNote. Queries are sent as parameters to filters through filter URLs and query results (triple tags and URLs of web pages) are sent back.



Input the query

<price, lessThan, 400>

Step 1 of 2: Configure the filters for your query

query	Select	filter	Description	Sample queries
<price, lessThan, 400>	☑	PriceFilter	find out the triples conveying the meaning "the price is less than a certain value"	<price,<, *>
	☐	BrandFilter	recognize triples of the form <brand, is var>	<brand,is, *>
	☐	ColorFilter	To find the triples that have a similar meaning as <color, is, deep>. (e.g. <color, is, dark>, <color, is black>).	<color,is, *> ,<color,=,*>

Step 2 of 2: Results

triple tags	Web page
<price, is, 300>	http://google.com/base/a/1677581/D10645289211991520771
<price, is, 290>	http://www.ohwuala.com/tho/tr/product/SONY-ERICSSON-W810I-WALKMAN-MOBILE-CELLULAR-PHONE-UNLOCKED--1001667.html
<price, is, 58 >	http://www.superwebshop.com/sonyericssont300.html

Figure 3. Triple querying takes place in two steps.

Triple querying takes place in two steps. Firstly, filters are selected and ranked. They are presented to users as shown in the right-up part of figure 3. It shows that a user issues a query <price, lessThan, 400> and the candidate filters are listed ordering by their commitments to the query. Then some filters (in our example, the first filter) are selected and by clicking the *next* button selected filters are triggered. Query results are shown in the left-bottom part of figure 3.

5 A Case Study

5.1 Case setup

In this section, we present a use case exhibiting how the triple tagging and collective querying work in the scenario of online shopping.

To this purpose, three filters are implemented. We first implemented a global *commitment* function for all filters based on their *sample queries*. Denote sample queries of a filter as $Q_s = \{q_1, q_1, \dots\}$ and the input query as q,

$$commitment_f(q) = \frac{1}{C}\min(Levenshtein(q_i, q))$$

where $q_i \in Q_s$ and C is the normalize constant. The sample queries of each filter are shown in the first row of table 2.

$Filter_1$ commits to recognize triples of the form <brand, is, var>, where *var* can be any string. Its *filtering* function are implemented as below,
$filering_{f_1}(G_{in}, q) = \{t|t \in G_{in}, \text{t = <brand,is, var'>}\}$
where *var'* is a substring of *var*.

$Filter_2$ commits to find out triples conveying the meaning "the price is less than a certain value". For a query q = <price, =, $\#$>, the filtering function returns a set of triple t, where t = <price, =, $\#'$>. Both $\#$ and $\#'$ are float numbers and $\#'$is less than $\#$.

$Filter_3$ commits to find triples that have a similar meaning as <color, is, deep>. Triples convey similar semantics also include, for example, <color, is, dark> and <color, =, deepgray>. For the purpose, a statistical (e.g. Naive Bayes) classifier can be trained a-priori.

5.2 A case study in Comparison Shopping

Suppose a user wants to buy a cellular phone. He starts from collecting information around the web. He tags the web pages of his interests in our system LegoNote. Three tagged web pages are shown in figure 1. Then he would like to make a comparison. We will show how the tripe query framework works in this case.

web page p_1	web page p_2	web page p_3
<price, is, 300>	<price, is, 290>	<price, is, 58>
<brand, is, Nokia>	<brand, is, Sony Ericsson>	<brand, is, SONY>
<color, is, silver>	<color, is, black>	<tax, is, 0>
<type, is, N95>	<shipping,fee,9.9>	

Table 1. Triple tags of three web pages p_1, p_2 and p_3.

	$filter_1$	$filter_2$	$filter_3$	
	brand, is,*	price, <,*	color, is, *	color, =, *
price, <, 100	9	**0**	9	10
price, lessThan, 400	14	**9**	14	15
brand, is, SONY	**0**	9	8	11
brand, =, SONY	**3**	11	11	8
color, is, dark	8	9	**0**	3
tax, is, 0	**6**	9	8	11

(The table shows the commitments of three filters to different queries given by the Levenshtein Distance. Results are shown without normalization.)

Table 2. Filter commitments.

First the user issues a query q = <price, <, 100>. Among the three filters in the system, $filter_1$ commits to the query the best (as shown in table 2) and is chosen. The triple tag <price, is, 58> is found by $filter_1$ and web page p_3 is returned.

Then a complex query Q = <price , lessThan, 400><brand, is, SONY> is issued. This time $filter_1$ is chosen for the sub query <price , lessThan, 400> and $filter_2$ is chosen for <brand, is, SONY>. The results returned by $filter_1$ are <price, is, 300>,<price, is 290> and <price, is, 58>. $Filter_2$ returns <brand, is, Sony Ericsson> and <brand, is, SONY>. Thus the combined result for Q is p_1 and p_2.

For the third time, the user tries another query on different attributes. Q=<color, is, dark><tax,is,0>. $Filter_3$ is chosen for the sub query <color, is, dark>. $Filter_1$ is incorrectly chosen for the sub query <tax, is, 0> and probably get truncated. Although there exists a triple <tax, is, 0>. So the result of Q equals to that of the query <color, is, dark>. $Filter_3$ is a classifier based filter and <color, is, black> might be found, at last web page p_2 is returned.

In the example the triple tags are added by the user himself. It is not necessary the case. The seller of the product or other users who are interested in the product can also add triples.

References

[1] Rich tags: Supporting better exploration of digital repositories with semantic social tagging, http://www.mspace.fm/projects/richtags/.

[2] D. Beckett. Semantics through the tag. XTech 2006, May 2006.

[3] P. Cudre-Mauroux, K. Aberer, A. Abdelmoty, T. Catarci, E. Damiani, A. Illaramendi, M. Jarrar, R. Meersman, E. Neuhold, C. Parent, K. Sattler, M. Scannapieco, S. Spaccapietra, P. Spyns, and G. de Tre. Viewpoints on emergent semantics. *Journal on Data Semantics VI*, pages 1–27, 2006.

[4] S. Golder and B. A. Huberman. The structure of collaborative tagging systems. *Journal of Information Science*, 2006.

[5] M. Guy and E. Tonkin. Tidying up tags? *D-Lib Magazine*, 2006.

[6] A. Marchetti, M. Tesconi, F. Ronzanona, M. Rosella, and S. Minutoli. Semkey: A semantic collaborative tagging system. In *WWW2007 workshop*, 2007.

[7] P. Mika. Flink: Semantic web technology for the extraction and analysis of social networks. *Journal of Web Semantics*, 3(2), 2005.

[8] X. Wu, L. Zhang, and Y. Yu. Exploring social annotations for the semantic web. In *Proceedings of the 15th International Conference on World Wide Web*, 2006.

[9] J. Yang and M. Ishizuka. Social graphic tagging for semantic metadata and a case study on consensus discovery. In *IJCAI-07 Workshop on Semantic Web for Collaborative Knowledge Acquisition*, Hyderabad, India, 2007.