Abstract. Collaborative tagging is increasingly drawing attention as a promising source of taxonomic information; it is often referred to as folksonomy. A tag is information that is added to a resource (such as pictures, a web page, or blog entries); a tag can be represented as an RDF statement. However, no explicit study has been undertaken to bridge the gap between annotated tags and RDF statements. This paper proposes a model to represent tag information using RDF statements. Particularly, we give an incremental extension path from keyword tagging, which we call single-tagging, to relational tagging, which we call double-tagging, finally to subject-predicate-object tagging called triple-tagging, which (theoretically) corresponds to RDF statements. We build a new tagging system called LegoNote as the first triple-tagging system, and report some preliminary experiments. Our work will clarify the natural direction from the current tagging system and provide insights to realize a full Semantic Web vision.

1 Introduction

Collaborative tagging is a process by which users attach metadata in the form of terms (or tags) to web resources (such as pictures, web pages, blog entries, and video clips) and share them socially. Tags are chosen freely by users, not from a controlled vocabulary. Using tags, a user can not only organize personal data, but can also browse and search the shared information efficiently [5]. The folksonomy, a portmanteau word combining “folk” and “taxonomy,” is the taxonomy generated by the shared tags. Already, many collaborative tagging systems are in use (e.g., del.icio.us, flickr, and CiteULike).

Collaborative tagging is appealing to the Semantic Web community because it suggests a promising, bottom-up approach to generate semantic metadata for web resources: Users can annotate tags freely; collective data will create the relations and hierarchy among tags [16]. Guy and Tonkin report that, in a collaborative tagging system, the use of the tag converges. For that reason, it is possible to obtain agreed emergent semantics [7]. Various computational models are proposed as well: Mika proposes a tripartite model that extracts domain ontology from tags by which tags related to the same domain are connected and extracted [15]. Steels and Hanappe strive to achieve emergent semantics on the use of labels between agents where agents negotiate the semantics of labels [17].

Some discussions discriminate collaborative tagging from folksonomy [14], but we use them interchangeably in cases where their use is not confusing.

On the other hand, numerous studies have explored the utilization of ontologies that are defined formally and explicitly by domain experts and ontology engineers. In this approach, metadata are generated either by the content creator or by ex-ante annotation. Both manual [9, 8] and (semi-) automatic [11,4,10] annotation methods are explored. This approach works best with the ontology engineer in the anticipated domain, but it is challenged when working with normal users across the Web because: first, the predefined ontology is not sufficiently flexible to adapt to diverse situations; second, the start-up cost is too high for a typical web user to investigate and use the ontology.

Some discussions have elucidated folksonomies and elaborately constructed ontologies (e.g., [3]). In his ISWC2006 speech [6], Tom Gruber pointed out the difference of folksonomy and ontology: ontologies are not taxonomies, they are for sharing, not finding. They enable cross-application aggregation and value-added services. We are aware of the power of folksonomy as well as the importance of ontologies. Therefore, the time is right for expanding the taxonomies given by current tagging systems toward more-structured ontologies using advanced tagging techniques. Compared to the Semantic Web initiate [2], where the metadata that describe web objects take the form of RDF triple, tags are still viewed as the set of keywords attached to a web resource. To render the emergent taxonomies as more structured, it is important to clarify the gap between current tags and RDF triples, as well as the path between them.

This paper proposes a triple-tagging scheme by which users can tag (i.e. annotate) web resources with RDF triples. The scheme not only incorporates the expressiveness, but also retains the flexibility of tagging by allowing users to choose their own terms in annotation. We argue that triple-tagging is a natural extension of the current keyword tagging by mapping them to a subject-predicate-object triple. Current keyword tagging allows users to input only objects. Therefore, we call it *single-tagging*. Some advanced tagging systems as well as some studies have been designed to allow users to input predicates (relations) in addition to objects. For that reason, we call the tagging *double-tagging*. Finally, we might have freedom to input all subject-predicate-object triples, which corresponds exactly to RDF statements. This freedom creates enormous potential because triple-tagging allows us to write about anything that is not the tagged web resource itself. As a tradeoff, it engenders some problems: How can we identify a subject uniquely? Do users really use such a tagging system?

We implemented a triple-tagging system called *LegoNote* to investigate the possibility of triple-tagging. Using LegoNote, users can make triple annotations to a web page and share the triples. Preliminary experiments were made. They revealed the merits and problems of triple-tagging. We discuss those results in addition to some points of advice for more-advanced triple-tagging systems in the future.

The contributions of the paper are summarized as follows:

- We propose sequential models from the current tagging to the RDF model, namely single-tagging, double-tagging, and triple-tagging. The models show us how we can gradually move tagging data toward RDF data.
- As one implementation of triple-tagging, we describe *LegoNote*, which enables users to make triple annotations to a web page.
- Discussions are presented relative to preliminary experiments to bridge folksonomy seamlessly with the Semantic Web for future triple-tagging systems.
Fig. 1. An example of single-tagging by del.icio.us.

The remainder of this paper is organized as follows. In section 2 we describe how triple-tagging augments the expressibility of current tagging systems through analyses of three tagging models. Furthermore, we explain how the triple-tagging scheme supports generation of the RDF triple for the Semantic Web. In section 3 the formal model of triple-tagging is introduced. The implementation of LegoNote is introduced in section 4, a case study is carried out and results of analyses are given. In section 6 we present possible approaches to provide users immediate benefit. Finally, the conclusion is explained in section 7.

2 Single Tag, Double Tag, and Triple Tag

In this section, we describe how the current tag patterns fit into the subject-predicate-object scheme and describe an incremental extension path from the keyword-based tag to the triple-tag through sequential models: single-tagging, double-tagging, and triple-tagging. In addition, we explain how the triple-tag scheme supports generation of the RDF triple of the web content. The triple-tag model acts as a bridge that leverages the current tagging data and the RDF metadata.

2.1 Single-tagging

In most current popular tagging systems such as del.icio.us and Flickr, web resources are annotated with tags, or simple keywords. We designate the tagging by which web resources are annotated using keywords as single-tagging. Figure 1 shows an example: a tagged web page in del.icio.us where four tags are added to the article Using iChat AV with a firewall or NAT router⁵. The four tags are todo, osx, macosx, and useful. As Golder pointed out [5], several types of tag can be added, such as identifying what the contents are about, identifying what it is, identifying qualities or characteristics, task organizing, refining categories, and so on. In this example, osx and macosx describe contents, todo is task organizing, and useful is identifying qualities.

Although some types are of tags, tags usually only vaguely describe the web resource contents. In other words, we do not know any concrete relation between the web resource and tags, except that the tags are relevant “in some way.” If we attempt to express the tagging information with the subject-predicate-object scheme [1], we can write as

\[(\text{uri of web resource}, \text{relevant-to}, \text{tag})\]

⁵ The URL is http://docs.info.apple.com/jarticle.html?artnum=93208-en
<item
  rdf:about="http://docs.info.apple.com/article.html?artnum=93208">
  <title>Using iChat AV with a firewall or NAT router</title>
  <link>http://docs.info.apple.com/article.html?artnum=93208</link>
  <description />
  <dc:creator>JDALaRose</dc:creator>
  <dc:subject>macosx osx todo useful</dc:subject>
  <taxo:topics>
    <rdf:Bag>
      <rdf:li resource="http://del.icio.us/tag/todo" />
      <rdf:li resource="http://del.icio.us/tag/osx" />
      <rdf:li resource="http://del.icio.us/tag/macosx" />
      <rdf:li resource="http://del.icio.us/tag/useful" />
    </rdf:Bag>
  </taxo:topics>
</item>

Fig. 2. RDF statement of tags by del.icio.us.

where uri of web resource is a subject which represents the uri of the web resource, tag is an object representing a tag, and relevant-to is a general property (which we define) that represents the relation between tag and uri, used as a predicate.

Consequently, we obtain four triples of the following form.

(http://docs.info.apple.com/article.html?artnum=93208-en, relevant-to, todo)
(http://docs.info.apple.com/article.html?artnum=93208-en, relevant-to, osx)
(http://docs.info.apple.com/article.html?artnum=93208-en, relevant-to, macosx)
(http://docs.info.apple.com/article.html?artnum=93208-en, relevant-to, useful)

Actually, some tagging systems generate such RDF statements. Figure 2 shows the RDF statements generated by del.icio.us. Although it uses the <taxo:topics> property instead of relevant-to and <rdf:Bag> to combine tags, the essential framework is identical: single-tagging generates RDF statements in which the objects are tags.

2.2 Double-tagging

As a natural extension, we consider double-tagging, where users assign the values of not just the object but also the predicate. It is readily apparent that tagging users are using double-tagging patterns in the existing folksonomy systems: as we described, the type of tag can be considered as a property (or a relation) of tags, such as (uri, what-it-is-about, osx). The usage of multiple tags sometimes can be understood as double-tagging. For example, in flickr, users tag the picture using color and white to express that the color property of the picture is white.

In this case, users can be considered as annotating a web resource using a triple.

(uri of web resource, property, tag)
In the examples given above, *color* and *what-it-is-about* act as the *predicate*, and *white* and *macosx* as the *object*. The *subject* (uri of the web resource) is assigned automatically by the system. Therefore, a user has the freedom to choose *predicate* and *object*.

A semantic collaborative tagging system, SemKey [12], is an example of a system that embodies the idea of double-tagging. In fact, SemKey proposes formulation of current tagging with *semantic assertions* by which the web resource is connected by a specified relation to a concept. The concept has different lexical forms which correspond to the keyword based tags, which might be of synonyms, alternate spellings or misspelling errors and all other possible lexical forms. Three relations are predefined: hasAsTopic, hasAsKind, and myOpinionIs.

A comparable approach is carried out by Google Base. With Google Base, users post their own items for advertising. Various items can be posted, e.g., housing information, events, products, reviews, and personal ads. Users are allowed to attach the user defined property-value pairs of the item when posting an item. Aside from the suggested property, users are free to add other information. Although Google Base is not a typical tagging system in that the metadata are created by the owner of the item instead of other users, it fits into our scope of discussion because Google Base users contribute metadata freely, without having to follow a predefined vocabulary. There are some other systems that can be considered as examples of double-tagging.

With double-tags, information is presumed to be retrieved more precisely. For example, if the user seeks pictures using the single tag *white*, the result might be a picture with a white flower, or a person with the name *White*. Using double-tagging, that confusion is avoidable using property names *color (=white)* or *name (= white)*, although it will burden users to make more annotations.

### 2.3 Triple-tagging

Finally, we describe the *triple-tagging*, which is a full version of tagging to web resources. Single-tagging and double-tagging can be considered as restricted versions of triple-tagging.

Let us assume that we have one web page of a new cellphone. We might want to tag this web page by say, *cellphone*, *cool*, and *Sony Ericsson*. (In double-tagging, we might want to tag *(uri, what-it-is, cellphone)*, *(uri, identifying-quality, cool)* and *(uri, what-it-is-about, Sony Ericsson)*). In single-tagging and double-tagging, we can make tags to the product, but we can not put any information related to Sony Ericsson. That is sometimes a large constraint when we want to put more detailed information for further utilization or aggregation of tagged items, not just for simple navigation. For example, if we can write that Sony Ericsson is owned by Sony and Ericsson, and that Sony is a Japanese company, we might list up the products when a user wants to see cellular telephones manufactured by a Japanese company.

In this case, we can make the subject-predicate-object triple as

\[(uri, property, tag)\].
The uri can be any uri of a web resource. Particularly, if the uri is the tagged web resource itself, the triple corresponds to that by double-tagging. Furthermore, if the property is restricted to relevant-to, the triple-tagging corresponds to single-tagging.

In triple-tagging, we can write a statement about Sonny Ericsson such as

\[(Sony\ Ericsson, \ owned-by, \ Sony)\ (Sony, \ is-a, \ Japanese\ Company)\]

To generate RDF statements, we must have uris for Sony Ericsson and Sony, and definitions of owned-by and is-a properties, as described later.

In summary, by triple-tagging:

- a user can create a triple statement about the (target) web resource e.g., (uri of cell phone, brand, Sony Ericsson); and
- a user can make a statement about web resources that are not the target web resource itself. e.g., (Sony Ericsson, belongto, Sony) and (Sony, is-a, Japanese Company).

Triple-tagging greatly enhances the expression of contents because we can say anything about anything on any web page (as long as the tagging is appropriate). It might seem too expressive as a tagging system, but it corresponds exactly to what we can achieve using RDF statements. By harnessing RDF statements, we can realize various kinds of retrieval and inferences thanks to many studies of Semantic Web data processing.

2.4 Triple-tags and RDF statements

Triple-tagging is a natural extension of the current tagging mode; some problems pertain when considering the generation of RDF statements.

A salient problem is how to assign a uri to each tag that should be a subject. Current tagging systems give a particular uri to a tag. Denote the prefix URL of the implementation system as system, the above triple (Sony Ericsson, owned-by, Sony) turns into the RDF triple of the following form.

\[(system:Sony\ Ericsson, system:owned-by, system:Sony)\]

However, to maximize interoperability, it is essential to connect the uri of the local system to a popularly-used uri. No such problem (at least superficially) pertains for single-tagging and double-tagging because annotated tags (or objects) can be treated as literals: local uris would work well. Another problem is that users would not produce a triple-tag if a system were to require too much formality. It is important for users to freely describe tags to make use of the concepts of Web2.0.

We have several solutions to balance the ease and formality of triple-tagging. First, we can compel a user to identify a uri whenever a tag is used as a subject. It restricts users from freely annotating triple-tags, although the obtained tags generate useful RDF statements. A second approach is to let users describe triples (without uri identification) first. Then, automatic processing to assign a tag to a uri is useful. For example, a triple-tag (Sony Ericsson, owned-by, Sony) can be set as an alias of the following RDF statement, of which the semantics of the word ship and time are defined in WordNet:

\[(wiki:Sony\ Ericsson, wn20instances:word-owned-by, wiki:Sony),\]
VII

where wn20instances is the prefix of the WordNet namespace\(^8\) and wiki is the prefix of the Wikipedia namespace\(^9\). The third approach enables users to input uri of a tag when they want to disambiguate it. As long as no difficulty arises, the system can use subjects of triples which have appropriate uris and those that do not. This is a very loose setting of the system, but it might work well when many users apply many tags.

We can discuss different settings for different purposes. For this study, because we seek to minimize users’ burdens, we employ the third approach. We will ignore the formality of triple-tagging first, and then let the user describe a uri only when she/he wants. In this sense, a triple-tag does not correspond to an useful RDF statement because subjects and predicates might have only locally-defined uri. For that reason, we express a triple-tag as an RDF-like statement to emphasize the difference. (Hereinafter, we use <> instead of () to express a triple-tag.)

3 Formal Model of Triple Tagging

A triple-tag is an RDF-like statement that consists of three ordered terms (keyword or combination of keywords) and takes the form of \(<term1, term2, term3>\). The semantics of the triple resemble those of the RDF triple \((subject, predicate, object)\). Users who triple-tag the web resource are presumed to accept the implied meaning of the triple-tagging scheme. For example, the triple \(<ship, time, 2days>\) is interpreted as “the time for shipping is 2 days”\(^10\). Other than that, no special requirement applies to the use of the triple-tag because we choose a loose setting to minimize users’ costs.

Let \(W = \{w_1, w_2, w_3, \ldots, w_n\}\) be a set of terms. The formal definitions of the single- / double- / triple-tagging are given below.

**Definition 1 (Single-tag).** A single-tag to web resource \(u_t\) annotated by term \(w\) is written as \(t_{\text{single}} = <u_t, p_{\text{rel}}, w>\), where \(p_{\text{rel}}\) is a property that represents any relevance of \(w\) to \(u_t\). It corresponds to an RDF statement \((u_t, p_{\text{rel}}, w)\) under which \(u_t\) and \(p_{\text{rel}}\) are properly defined. Single-tagging is to annotate \(u_t\) with \(t_{\text{single}}\).

**Definition 2 (Double-tag).** A double-tag to web resource \(u_t\) annotated by term \(w\) with property \(p\) is written as \(t_{\text{double}} = <u_t, p, w>\). It corresponds to an RDF statement \((u_t, p, w)\) under which \(u_t\) and \(p\) are properly defined. Double-tagging is to annotate \(u_t\) with \(t_{\text{double}}\).

**Definition 3 (Triple-tag).** A triple-tag to web resource \(u_t\) on web resource \(u\) annotated by term \(w\) with property \(p\) is written as \(t = <u, p, w>\). It corresponds to an RDF statement \((u, p, w)\) under which \(u\) and \(p\) are properly defined. Triple-tagging is to annotate \(u_t\) with \(t\), where \(u\) can be the same as \(u_t\).

We can clearly understand the constraints on single- and double-tagging compared to triple-tagging. Next, we will define a tag graph, that (potentially) corresponds to an RDF graph. We will show the tag graph obtained by our LegoNote in Section 5.

\(^8\) http://www.w3.org/2006/03/wn/wn20/instances/
\(^9\) http://en.wikipedia.org/wiki/
\(^10\) It might be written more precisely as \(<\text{uri}, \text{shiptime}, 2\text{days}>\), but we permit any triples.
Definition 4 (Tag graph). Given a set of triple-tags, $T$. For each $t = <u, p, w> \in T$, we set $u$ and $w$ as nodes, and we set an edge from $u$ to $w$ (annotated by $p$). We call the resultant graph as a tag graph $G = (V, E)$, where $V$ represents a set of nodes and $E$ represents a set of edges.

Usually tag data is represented as $(U, W, I)$ where $U$ is the set of users, $W$ is the set of tags, and $I$ is the set of instances (web resources) [16]. We can also consider that the triple-tag data is represented by $(U, W, I, R)$ where $R$, the relation among tags, are added newly. Though we should define various dimensions more precisely in the future, we do not explain further here in order to avoid complexity. Rather, we would explain the implementation and results to reveal the utility of triple-tagging empirically.

4 Triple Tagging System: LegoNote

4.1 Creating triple-tags

LegoNote\(^{11}\) is an implementation of the triple-tagging scheme. Using LegoNote, users can create triple-tags for the web page and share them by saving them to the LegoNote server. The tag words are chosen freely. The shared triple-tags will be shown as well when the user is tagging a web page that has been tagged before by others. In LegoNote, we present triple-tags graphically. The graphic presentation is easy to understand and intuitive to users. In the following, we will introduce the system interface and its implementation in detail.

Figure 3 is a screen shot of LegoNote depicting how a user triple-tags a web page in LegoNote. The left-hand side of figure 3 is an embedded browser, which occupies most of the space to reserve the familiar browsing experience for the user. The user can browse web pages either by entering a URL to the embedded browser or by following the web page links.

The graphic tagging tool is floating on the right-hand side. The upper area is the tagging window, in which users can create triple-tags graphically. Basic operations are supported, showing themselves as links in the toolbar (AddNode, AddEdge, Move, Edit, etc.). To create a triple-tag in LegoNote, the user can first add two nodes, and then connect them with an edge. Names of the triple element (the node or the edge) can be edited. The user can delete the triple element by first mouse-selecting it and then pressing the “Delete” key. In addition, the nodes can be dragged around for a better layout.

The user can tag either the whole document or a document fragment. For the former, a user simply adds triples in the tagging window. For the latter, the user first selects a piece of text and then creates a triple. When the triple is created, if any selected fragment of the web page exists in the embedded browser, the system will record its content and location for the triple. When we click a tag element, the fragment will be highlighted and located automatically if it is connected to a fragment of the current document.

The tagging window of figure 3 shows triple-tags of user Tony, where the following triples are created:

\(^{11}\) The online demo is accessible from http://dom-sensus.sourceforge.net/
(A web page about a cellular phone is loaded in the embedded browser. The user’s triple-tags are shown as a graph in the upper right area. The same page is also tagged by two other users: Frank and Yang. Yang’s tags are also loaded and shown in the figure.)

**Fig. 3.** How a user tags a page using LegoNote.

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

where we use - indicating that the edge name is not given.

Triple-tags can be uploaded to the server so that they are accessible to others. The bottom area of the floating tagging tool, which we call the referencing window, exhibits the triple-tags of the same page shared by other users. It helps the current user grasp the important information quickly and build his own graph easily. As shown in figure 3, a web page about a cellular phone Sony Ericsson K310i is loaded in the embedded browser, aside from the current user Tony, two users, Frank and Yang, also tagged the same page and shared their triple-tags. Yang’s tags are shown.

### 4.2 System architecture

LegoNote is implemented as an AJAX application with Browser-Server architecture. Figure 4 depicts the overall architecture of LegoNote. On the client side, a Scalable Vector Graphics (SVG) graphic library is implemented. The floating toolbar for graphic

---

12 http://www.w3.org/TR/SVG/
tagging and a tag browser is built using the library. The graphic tagging toolbar interacts with the embedded browser by manipulating the DOM\(^{13}\) of the web page. Instead of requesting pages to the Web directly, all the HTTP requests initiated from the embedded browser are redirected to a proxy server. The proxy server then initiates the request, fetches the page from the Web and sends the returned page content back to the embedded browser in the form of JavaScript Object Notation (JSON). Communication between the embedded browser and the proxy server is carried out using EX-AJAX\(^{14}\). This additional process is necessary to bypass the “same domain” security restriction. Otherwise, the tagging toolbar is forbidden to manipulate the web-page DOM.

On the server side, a Tomcat application server runs to possess the user requests, such as user logging-in, saving, and fetching the triple-tagging data. Sesame\(^{15}\) manages the storage and querying of the triple-tag that is modeled by the triple-tagging scheme.

5 Preliminary Experiment

We conducted a small experiment to obtain feedback on the utility of triple-tagging. Four college students (other than the authors) were invited to our experiments. We sought to find out the features of the triple-tags, and whether triple-tagging engenders different metadata from the keyword based tagging and how they mutually differ, as an indication for in-depth work.

The experiment is set as follows. First, we prepared a data set by downloading 300 web pages from \url{http://www.pconline.com.cn/mobile/review/} where people write their reviews about mobile phone. Then a video clip was made, instructing participants in how to use the graphic tagging toolbar and demonstrating a constructed tag graph on

\(^{13}\) \url{http://www.w3.org/DOM}  
\(^{14}\) \url{http://ajaxextended.com/}  
\(^{15}\) \url{http://www.openrdf.org/about.jsp}
a researcher’s home page. The demonstrated tag graph is related to a totally different domain because we tried to avoid imposing too much bias on the testers. Finally, students who had experiences with current tagging systems but who knew nothing about the Semantic Web and RDF were invited to join the experiment. First, they were asked to single-tag some web pages. Then after learning how to use the graphic tagging tool from the video, they are asked to tag the same web pages and to write down some comments. The web pages were selected randomly from the dataset.

In the experiment, there are 11 different web pages tagged by the four users. Altogether, 104 nodes and 88 edges were created. The words (separated by blank space) used for nodes were 202; those used for edges were 112. For all the figures above, duplicated elements were counted separately. The observations are described below.

First, the result shows visually that users are willing to triple-tag the web resource when it is possible. Because the number of the edges (88) resembles that of the node (104), however, quite a few edges (50 out of 88) are left unnamed, which suggests that users are most interested in adding information that is directly useful to them. In the triple-tagging scheme, the node name is the most informative part; the edge helps to organize nodes. The edge name however, is not as informative as the former from the tagging user’s point of view.

The second observation is that, compared to the keyword-based tagging scheme, triple-tags provide a more detailed description of the domain. That advantage is attributable to the feature of the triple scheme, which allows the user to insert statements into the web content. Table 1 and Table 2 respectively show lists of the words used for triple-tagging and single-tagging. In those two tables, words are listed in groups.

Table 1. Vocabularies used for triple-tagging.

<table>
<thead>
<tr>
<th>Frequency</th>
<th>Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>65536, 8, 40, 686, 2002, 4096, 8280, 17911, 17951, 200000, #3, #4, 116 mm<em>20 mm, 116 mm</em>57.5 mm<em>20 mm, 128</em>128, 130 g, 160 g, 1 g, 206 Mhz, 240*320, 260 color, 2-monitor, 3.5&quot;, #5536 color</td>
</tr>
<tr>
<td>2</td>
<td>2.9, 32 Mb, battery, book, business, color, colored, components, display, electronic, evaluation, focus, handy, inch, input, manufacturer, memory, model, new, Nokia, other, pc, pixel, productions, rom, Samsung, smart, software, sound, super</td>
</tr>
<tr>
<td>3</td>
<td>camera, category, keyboard, physical, resolution, weight</td>
</tr>
<tr>
<td>4</td>
<td>and, design, fit, pocket, screen, volume</td>
</tr>
<tr>
<td>5</td>
<td>#1, #2, cos, mobile</td>
</tr>
<tr>
<td>7</td>
<td>function, phone</td>
</tr>
</tbody>
</table>

Table 2. Vocabularies used for single-tagging.

<table>
<thead>
<tr>
<th>Frequency</th>
<th>Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>a10000, abundant, accessory, address, adjustable, appearance, background, basic, body, cameras, card, cell, cellphone, cheap, Chinese, chord, circular, cpu, display, dopod, double, editing, effect, electronics, embedded, excel, external, fabricate, flash, functions, game, games, gm958, good, gprs, high, identification, intel, java, lcd, mail, main, marvelous, media, message, messages, microsoft, mode, more, Motorola, MSN, night, no, non-pnp, operating, operation, os, panda, picture, player, price, primary, properties, ram, recording, ring, s320, screen, sdom, shooting, short, signal, single, space, strongarm, symbain, system, t9, tiny, two, video, voice, weak, weaker, weight, word</td>
</tr>
<tr>
<td>2</td>
<td>2.9, 32 Mb, battery, book, business, color, colored, components, display, electronic, evaluation, focus, handy, inch, input, manufacturer, memory, model, new, Nokia, other, pc, pixel, productions, rom, Samsung, smart, software, sound, super</td>
</tr>
<tr>
<td>3</td>
<td>camera, category, keyboard, physical, resolution, weight</td>
</tr>
<tr>
<td>4</td>
<td>and, design, fit, pocket, screen, volume</td>
</tr>
<tr>
<td>5</td>
<td>#1, #2, cos, mobile</td>
</tr>
<tr>
<td>7</td>
<td>function, phone</td>
</tr>
</tbody>
</table>
Table 2. Vocabularies used for single tagging.

<table>
<thead>
<tr>
<th>Frequency</th>
<th>Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3650 a1000 benz big camera cdma CECT customization display double</td>
</tr>
<tr>
<td></td>
<td>evaluation eZi female functions gm958 highPixel informatic-product</td>
</tr>
<tr>
<td></td>
<td>keyboard Korea LG8280 local mobile-phone mobilephone motorola</td>
</tr>
<tr>
<td></td>
<td>multiply nokia P800 P858 panda S320 sumsungSGH x209</td>
</tr>
<tr>
<td>2</td>
<td>sumsung, Smartphone, mobile, function, dopod</td>
</tr>
<tr>
<td>4</td>
<td>phone</td>
</tr>
</tbody>
</table>

by their frequencies of occurrence. Compared to the single-tagging model, more words related to the domain (cellular phone) are used in the triple-tagging scheme, e.g., memory, sound, battery, and so on. The more general the concept described by the word, the more times the word appears in the results. The words function and phone which are the most general concepts, appear the highest seven times. Note that in table 1, words that have frequency of one are further divided into groups. Those in the upper group are related to information that is unique to the resource. They are presumed to be rare and the frequency is low. Information of this group rarely appears in single-tagging systems.

The third observation is that triples vary greatly. We connect the triple-tags of the 11 web pages with nodes holding the same name, thereby forming a summarized triple-tag graph, which is shown in figure 5. As described above, users tend to connect the nodes when creating their own tag graphs. However, there are not so many interconnections among graphs shown in figure 5. Most of the time, the tag graph of a web page stands alone and only occasionally connects to others using nodes of the same name. In fact, we failed to find even a pair of triple tags that are the same. The scarcity of experimental data partially explains the phenomenon, but it might be a crucial problem for future studies.

In addition to creating triple tags during the experiment, users were also asked to write down some comments about the triple-tagging. Two comments are especially noteworthy: The first that the graphic tagging tool was interesting, but not convenient. The second comment is that functions such as triple-based querying should be provided so that users can explore information more efficiently.

6 Discussion

This experiment was small; it provided no quantitative proof of the utility of triple-tagging. Nevertheless, results of the experiment indicate that users will triple-tag a web page when useful tools are provided. Furthermore, triple-tags reveal many more domain details than single tags do.

The challenges revealed by the results of this experiment are as follows: to describe something in detail, instead of endeavoring to build triples, users tend to use phrases as the name of a node that can be split into triples. A possible solution is to extract triples (semi-)automatically from the phrases that are chosen (or input) by the user.

As described in Section 2.3, the triple-tag scheme serves two roles: to describe the web document contents and to connect and organize existing keyword tags. In this experiment, the second role of the triple-tag does not appear much, which indicates that the scenario and methods for the two types should be different. We can apply our triple-tagging model to entity relations that are extracted using a Web mining approach [13,
Fig. 5. Summarized tag graph.

The relations among entities extracted by Web mining correspond to triple-tags in our system. The generated RDF statements will enrich the RDF data significantly if we can assign a uri to each named entity (automatically or manually using systems like LegoNote).

The variation of triple-tags reminds us that we should provide methods that accelerate triple converging process and which provide instant feedback to the user as local instant feedback instead of global instant feedback.

We presented that a triple-tag is (ideally) an RDF triple. For that reason, query mechanisms such as SPARQL\(^\text{16}\) and Sesame\(^\text{17}\) can be explored. A problem is that without an *a priori* agreement between the triple creator and the query issuer, it is difficult to construct the appropriate queries that retrieve triple-tags efficiently. A possible solution is to map triple-tags into the RDF statement that uses the predefined vocabulary. Consequently, the triple will be recognized by the query that commits to the same vocabulary. In the solution, mappings from triple-tags to RDF statements should first be carried out before they are queried. The solution is not preferable because it offends the essential requirement of social tagging system: to provide instant benefit. This kind of mapping should be carried out after a huge number of triple tags are produced.

7 Conclusion

In this paper, we introduced a triple-tagging model, which is a natural extension of the keyword-based current tagging. Its definition is given formally. We argue that it serves as a bridge connecting social metadata and the machine-understandable semantic metadata of web resources. We implemented a system LegoNote for triple-tagging. Preliminary experiments show us the merits and problems of triple-tagging, which creates

\(^{16}\) http://www.w3.org/TR/rdf-sparql-query/
\(^{17}\) http://www.openrdf.org/
a further-developed triple-tagging system for using tagging data and also web mining data for the Semantic Web. We hope that our model and experiment will provide useful insights to bridge Semantic Web and folksonomy.

References