

Using Ontologies to Strengthen Folksonomies and Enrich Information Retrieval in Weblogs

Theoretical background and corporate use-case

Alexandre Passant
LaLICC, Université Paris IV
28 rue Serpente
75006 Paris, France

alexandre.passant@paris4.sorbonne.fr

&

Electricité de France, R&D
1, Avenue du Général de Gaulle
92141 Clamart, France

alexandre.passant@edf.fr

Abstract

While free-tagging classification is widely used in social software implementations and especially in weblogs, it raises various issues regarding information retrieval. In this paper, we describe an approach that mixes folksonomies and semantic web technologies in order to solve some of these problems, and to enrich information retrieval capabilities among blog posts.

We first introduce the corporate context of the study and the issues we have faced that motivated our approach. Then, we argue how the use of domain ontologies combined with the SIOC vocabulary on the top of an existing folksonomy and weblogging platform offers a way to get rid of free-tagging classification flaws, and enhances information retrieval by suggesting related blog posts.

Aside of the theoretical background, this paper also focuses on implementation. We present experimental results of this approach through the example of add-ons to a corporate blogging platform and the associated semantic web search engine, that extensively uses RDF and other semantic web technologies to find appropriate information and suggest related posts.

Keywords

Semantic Web, corporate information system, folksonomies, ontologies, information retrieval, SIOC, knowledge management

Introduction

Within the context of an economic and technological monitoring project, EDF¹ Recherche et Développement (EDF R&D) has setup an internal blogging platform, in order to allow

¹ Electricité de France (also known as EDF) is the leading french company in energy domain. Its R&D division employs about 2000 people and is involved in various activities. See <http://retd.edf.fr> for more information.

its engineers and researchers globally exchange pieces of information that they were previously used to share mostly in private communities.

In order to index blog posts created within this platform, a practice of free-tagging has been introduced, leading to the emergence of a folksonomy [22], i.e. a collaborative and open classification technique driven by users themselves, that can freely evolve among time [11]. Yet, while the first year and half of this experiment was a real success in terms of blog posts creation, this practice quickly raised various problems regarding information retrieval.

Because of these issues, we realized that using basic free-tagging was not enough regarding our needs in terms of access to information.

This rest of paper is divided as follows: section 1 introduces the motivations of corporate blogging, and the limits of free-tagging we identified regarding information retrieval in our context. Section 2 describes how semantic web technologies were adopted in order to solve some of these problems, and how we mixed the user-friendly and collaborative concept of free-tagging and a formal description of a domain using different kind of ontologies [12], especially domain ontologies and the SIOC [3] vocabulary, designed to represent metadata about online communities. This section also begins with an overview of related work in the domain of semantic blogging. In Section 3, we detail the complete architecture of the system, from the way how RDF metadata is created and stored for each blog post to how our semantic search engine retrieves information using the SPARQL [20] query language in a user-friendly way. Then, in section 4, we present how the system has been improved in order to infer related posts for a given tag, and so enhance information retrieval experience.

In each section, we will consider both theoretical approach and implementation, as a lot of efforts have been made to provide as easy as possible interface for users of the platform.

Finally, we will conclude this paper with an overview of what have been exposed so far and possible directions for future works.

1. Corporate blogging and information retrieval

1.1 Motivations of corporate blogging

As well as in most corporate information systems, EDF R&D provides different tools to store, query and retrieve internal documents, using workflow procedures, templates, search engines, and so on. On the other hand, employees often spontaneously create and spread among the company a lot of useful piece of information, as news regarding some of the company's customers, new prototypes or software releases, or informal meeting minutes. Yet, this kind of information cannot be stored using the previously mentioned tools, as it mainly consists in small and unstructured chunks of data that are not in the scope of common corporate tools, that mainly deal with structured and complex documents, see Fig. 1.

Corporate information	Spontaneous information
Template	Unstructured content
Validation workflow	Instantaneously produced
Reusability	Volatile information
Global archives	Personal desktop
Corporate access rights	Restricted scope

Fig. 1: Characteristics of different types of information within a company

Thus, engineers or researchers used to exchange these pieces of information mainly through e-mails, or - even worse - only by informal conversations during their coffee-break or at lunch time. This common process of delivering information through e-mail was not efficient for different reasons:

- as it consists in a closed diffusion process, where recipients have been chosen by the author himself, people that have not been involved in it cannot react to the information, even if they are experts of the domain within the company or have additional knowledge to offer;
- except mailboxes search engines, there is most of the time not efficient system to retrieve or query exchanged information. Moreover this is only locally accessible;
- when people involved in the process leave the project or the company, information will be lost; on the other side, new researchers or engineers will not be able to access the previously exchanged information, as they have not been involved in it.

In order to solve these problems, the decision was made to experiment weblogging as a open medium for all of these data [10]. The main goals were to reduce e-mail dependency and to incite people to explicit their atomic knowledge to an open audience, instead of restricting it to some private communities only, by creating an architecture of participation inside the company. In order to make it as successful as possible, we insisted on the benefits of such a social approach in both writing and reading aspects: information providers can have more information from comments on their posts, sometimes from people they might did not know before, who can enrich real-life connections in the company; while readers could learn unforeseen information about some companies or projects they might be interested in regarding some part of their work.

At the beginning of our experiment, the system only included RSS subscription mechanisms, in order to let people to become familiar with this idea of spontaneous, short and

efficient information. A few weeks after, weblogging capabilities were added to the the platform in order to let RSS subscribers react to some piece of news, and then create their own ones by writing blog posts. We introduced the ability to blog from existing news items or from scratch, and also to comment blog posts or create new ones from existing, using trackbacks.

Moreover, we also included group wikis into the platform, which can be seen as a complete "Enterprise 2.0" [16] information system, introducing social software and collective intelligence in the company.

After one year and half of existence, more than 8000 posts have been created, with about 80 bloggers and more than 600 readers or feed subscribers.

1.2 Adding meta-data to blog posts

In order to efficiently index created data, and then to easily retrieve information, a common practice, especially on the web and in the blogging area, is to use metadata attached to each blog post. Thus, additional and descriptive information about the blog post is added to each item. Nevertheless, a distinction must be done between metadata that will represent (1) the document and (2) its content, i.e. distinguish metadata the identifies (1) the document itself (eg: date, author), and (2) the one related to what the post is about (eg: topics ...).

Regarding the description of the document, a common practice is to use Dublin Core [9] terms and its refinements to add librarian metadata to the post, as its creation date (using `dc:created`), title (`dc:title`) or author (`dc:author`). On our platform and as in most of weblog engines, this kind of metadata is produced by the system itself and do not require any user intervention.

Regarding the blog post content, there are two main different methods to add such metadata to a document:

- use a set of predefined categories (thanks to taxonomies, ontologies ...). Then, users have to choose between different categories to get the one(s) that best define the content of their document. This approach widespread in librarian systems;
- associate free-keywords - in both type and numbers - to documents. Users are free regarding the choice of keywords they can associate to documents. This use became popular these last years thanks to the emergence and adoption of social software and "Web 2.0" social services.

Even if the first approach was common for most of our users that have never used weblogging before, we decided to initiate a free-tagging classification as we wanted to see how a corporate context can react to such a popular web practice, and how it could be adopted, and adapted if needed.

1.3 Identifying limits of free-tagging

As previously mentioned, our first year of experiment with weblogging was a success in terms of productivity. Yet by using free-tagging classification, we rapidly realized that even if users adopt it pretty easily, it was not sufficient regarding our needs in information search and retrieval, for three main reasons we identified as the following:

- tags variation. Even if different tags have the same meaning in the users mind, nothing can help the platform to create connections between them. For example,

the system will not consider tag **semantic web** as similar to the tag **semweb**. As anyone can use his own tags, this is an issue we have been rapidly faced with. Moreover, typo errors are also an important source of such variations;

- tags ambiguity. On the other hand, a single tag can express various ideas, and once again the system will not be able to make any difference. This is especially true when dealing with acronyms, that may - in our case - represent different organizations. This problem can also be seen on the web, for example on **del.icio.us** with computer-related acronyms, see Fig. 2. Users can indeed find which item is associated to the searched tag, but this implies an additional effort, and can be difficult when an item title is ambivalent or when there is no other tag associated to the item;
- flat organization of the tags. This is not a problem related the tag notion itself, but to the structure of collaborative tagging, contrary to predefined schemes as taxonomies or ontologies. Hence, tag search engines offer no way to find, or even to suggest, - for example - posts tagged with **rdf** when looking for **Semantic Web** items, even if from a user point of view, it would be useful to look at it.

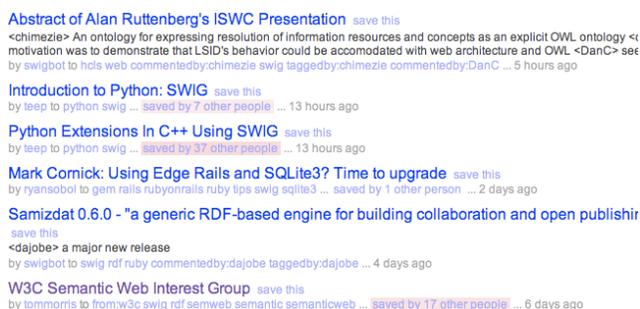


Fig. 2: *del.icio.us* results for **SWIG** tag: *C++ programming or Semantic Web* ?

Regarding tags variation and ambiguity, a former analysis lead us to identify various linguistic phenomena as the causes of these problems, and to offer some services to our users as tag completion or tag suggestions from blog post content, in order to reduce these flaws [19].

Yet, in order to offer better solutions to solve these variations and ambiguity problems - that are actually the reasons why controlled vocabularies as taxonomies have been introduced - and also offer a richer blog search experience to our users by suggesting related posts, we introduced a practice that kept the simplicity of free-tagging and added a formal layer on the top of the folksonomy, using semantic web technologies.

2. From folksonomies to semantic blogging

2.1 Related work

Mixing weblogs and semantic web [2] technologies, which goal is to offer better understandable data for machines, has already conducted to numerous works since the first mentions of semantic blogging, with various interests and motivations [6].

Thus, regarding software implementations, the Snippet Manager [7] uses semantic annotations to classify and share information snippets as blog posts, while semiBlog [17] integrates desktop applications and semantic annotations into weblogs.

In the domain of a global definition of semantics for the blogging activity itself, and more generally for online communities, one of the most recent work is the SIOC² ontology [3]. Its goal is to represent every objects and actions of an online community (users, posts, replies ...) as RDF [15] statements using this ontology in order to get uniform and shared representation format for any online community and create bridges between them. SIOC also allows to add metadata to blog posts, as it is strongly mapped to existing ontologies as Dublin Core or FOAF [5] for the people aspect.

Some recent efforts have also been done to embed semantics about blog posts content in the posts themselves. Thus, microformats³ provide an easy way to embed semantic information in a blog post by using HTML elements, but without the openness and extensibility that RDF offers. On the other hand, RDFa [1] initiative from W3C allows people to embed RDF in XHTML. In both case, one can use GRDDL [8] transformations in order to extract RDF data from these documents and use it as valuable information.

Regarding folksonomies and the semantic web, some approaches already defined ontologies of tagging: [18] defines a tag ontology with links as **relatedTag** or **equivalentTag** to define relationships between tags, while [13] focuses on the notion of tagging as a threefold action.

2.2 Linking tags to ontology concepts

While domain ontologies cannot be easily used in an open environment such as the web, since user consensus is needed regarding a particular model of the world, the corporate environment is a nice playground for this research area, as it is easier to agree on a view of a domain within the company.

Thus, our idea is to apply domain ontologies at the top of an existing folksonomy. By linking each tag to ontology class or instance - for example the **SWIG** tag can be related to the Semantic Web Interest Group -, we strengthen the folksonomy as we offer a way to remove ambiguity and get rid of tags variation, but also provide ways to infer related posts using properties defined in domain ontologies. Indeed, tags are linked to objects they represent contrary to [18] where **Tag** objects only link to keywords and consequently are not enough to be distinguished and deal with ambiguity.

So, in our approach, each tag can be associated to one or more ontology concepts, i.e. classes or instances - as in the rest of this paper. We introduced a basic RDFS [4] ontology with a **tag:hasTag** property to link any **rdf:Resource** to a tag. This way, we deal with tags variation - as exposed in section 1 - as we can link a single resource to various tags, see Fig. 3

Moreover, we do not consider tags as objects as in [13] or [18], but as simple text strings mainly because:

² <http://sioc-project.org>

³ <http://microformats.org>

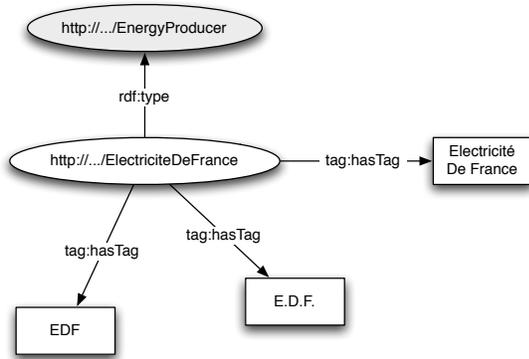


Fig. 3: *Linking various tags to ElectriciteDeFrance instance*

- in this approach, all metadata associated to tags (creation date, involved users ...) can be retrieved using SIOC and its properties, as this helps to create the threefold relation between people, tags and posts;
- we do not want to infer relationship at the tag level, but between posts and ontology concepts associated to the tags, so we do not need to add properties to tags.

Then, since we link tags to ontology classes or instances, in case of ambiguity, a tag can be associated to different ontology concepts. Thus, there is a need to create a direct link from the blog post to the right concept(s) to get rid of ambiguity. In order to semantically represent these connections, and also librarian metadata associated to blog posts, we decided to use the SIOC ontology, which provides an interesting framework to be used in this architecture for various reasons:

- as it provides an ontology for online communities, its scope is not limited to blog posts but covers relationships between posts and users, comments, and thus can be used as the core ontology of our socially-enabled information system;
- thanks to its mappings with different ontologies, we can keep the trace of original tags for a blog post using the `dc:subject` property, while `sioc:topic` will be used to link the post to corresponding ontology class or instance, since the goal of `sioc:topic` is to link any blog post (i.e. `sioc:Post`) to `rdf:Resource`.

Such associations allow to get a single tag expressing different ideas, but thanks to the ontology level, we can see - both from users and machines point of view - what is the meaning of a tag for a given post. For example, we can say that the EDF tag corresponds to the company instance `ElectriciteDeFrance` in a post, and to the french football team `EquipeDeFrance` in another one without ambiguity by combining the `dc:subject` and `sioc:topic` properties for a given post, see Fig. 4.

All these associations between posts, tags, and ontology concepts are created with simple user interfaces that will be described in the next subsection, and used in our semantic search engine that will be presented later.

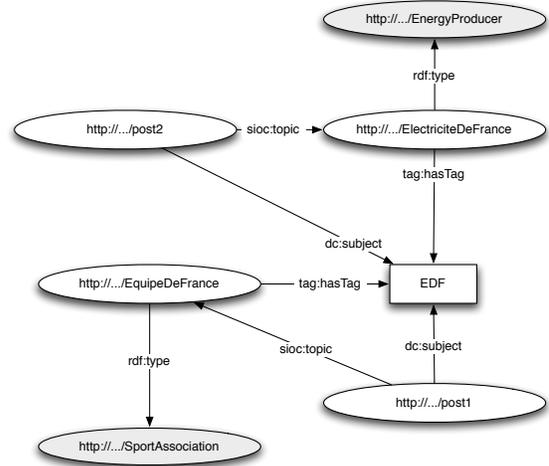


Fig. 4: *Solving ambiguity for the EDF tag*

2.3 User interfaces

In order to let users associate tags with ontology concepts for each post they write, we tried to make things as easy as possible. First, regarding the blog post creation and content metadata, users can still use free-tagging, by associating as many tags as needed to their post. Our completion system mentioned in 1.3 have been enhanced so that tags corresponding to existing ontology concepts are displayed in a different color than classical tags.

After validating the post and associated tags, users will be redirected to a page where they can check and validate relationship(s) between tags and ontology concepts for the current post, so that we can get rid of ambiguity, and also manage tag variation. User is then confronted to the following alternatives:

- if the tag is already associated to one or more concepts or instances, user will just have to check which associations are valid for this blog post. So, if he is using a tag that is linked to more than one concept, he will be able to remove ambiguity for the tag in this context, see Fig. 5;

"CNE" correspond à: _____

Commission Nationale de L'Energie (CNE) (Agence)

Contrat Nouvelle Embauche (CNE) (Loi)

[>> Vers l'interface avancée](#)

Fig. 5: *Checking links between tags and ontology instances*

- if the tag is not associated to any concept, the user will be able to browse the current ontology - and its instances - to link the tag to any existing class or instance. So, if a user choose a tag that represents a class or instance that already exists but that is not associated to it, for example using a new "eDF" tag for `ElectriciteDeFrance`, he will be able to link it to the

instance, so we can manage tag variation. Then each time a user will use this tag, he will get the newly-created association it in the previous step, adding a collaborative aspect to this step of defining tags for any concept thanks to the blogging community;

- if the tag represents a concept that is not in an ontology instance, user will have to contact the platform administrators so that they can add it. At the moment, we decided not to allow users to create instances, in order to avoid problems we have been faced with regarding tags variation.

2.4 Maintenance and usage of the ontology

As this approach deals with domain ontologies, we must think of their evolution and maintenance. Here, we must distinguish the ontology design itself, and creation of new instances.

Regarding the ontology, the scheme should not evolve a lot during its lifetime. Moreover, we wrote a set of modular ontologies, based on Proton [21], by designing and extending various modules thanks to analysis of hundred of tags and discussions with experts of various domains. These modules can be refined or adjusted in the future if needed. Regarding the instances, our system provides a back-office interface displaying the latest created tags, allowing platform administrators to create new concepts from these tags when needed, and to make needed associations. Concepts can also be created from scratch, especially when new bloggers join the platform and have previously asked for the creation of dedicated RSS feeds about some companies or technologies, which make us think they will blog about it.

As we deal with ontology instances, that have various properties depending on their type, we can add more information about them. Then, at the moment we automatically add geo-coordinates for each created instance of `Location` class (or its subclasses as `City`), using the `geonames.org` web-service. Thus, we offer users an alternate view of tagged posts, using a GoogleMaps mash-up, see Fig. 6.

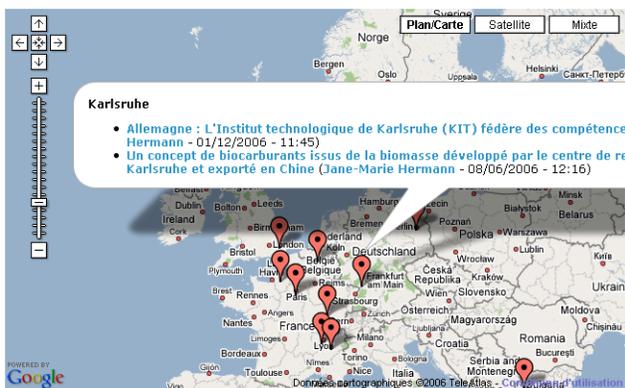


Fig. 6: Geolocation of blog posts using ontology instances

In the future, our back office will be completed so that users can add other properties to such concepts, while other part of our work could consist in extracting some of these properties from blog posts.

This process of ontology maintenance can be seen as a socially-driven ontology population approach since ontology and its instances evolve thanks to the tags used in the platform. Yet, we added a supervised aspect first in order to

avoid instances explosion (and get the same issues we have been faced with regarding the folksonomy), but also as we did not want our users to face with a defined classification, as we offered them an open-tagging mechanism through free keyword capabilities.

3. A semantic search engine

3.1 Global architecture

As previously mentioned, each post has an RDF representation using the SIOC ontology, which is created when the user saves his post. On the other hand, ontology and instances are created with other interfaces and consists in OWL/RDFS and RDF files.

Thus, in order to make queries over all this files, we decided to store all this data in a triple-store. We chose the three-store[14] engine, as it provides support for context - in order to keep information about provenance of each RDF statement - plus a nice API to add data, and also offer SPARQL [20] support, the leading semantic web query language for RDF and support for inference rules.

As blog posts are written independently, and in order to have a real-time synchronized RDF store, each post is pushed to the store as it is created or modified.

On the top of this RDF store, which is running on a MySQL based Linux server, we provided search interfaces to allow users retrieve blog posts, thus creating a complete semantic web infrastructure atop of our corporate blogging platform, see Fig 7.

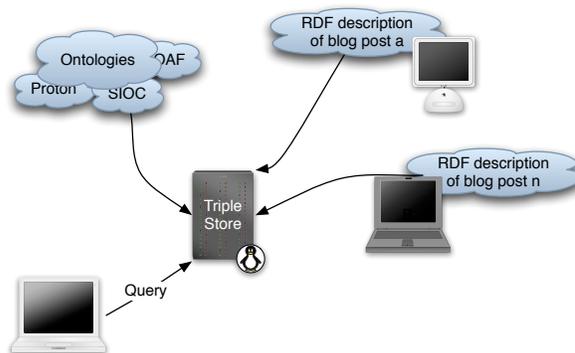


Fig. 7: Our semantic blogging architecture

3.2 Retrieving blog posts

When a user looks for blog posts related to a given keyword `$tag` - since the engine offers plain-text search only at the moment -, the system will retrieve all ontology concepts associated with this tag - or tags that contain the typed expression -, using the following SPARQL query snippet⁴:

```
SELECT ?concept
WHERE {
  ?concept tag:hasTag ?tag .
  FILTER(?tag, "$tag")
}
```

⁴ Please note that vocabulary prefixes are not mentioned in the examples of this paper

If more than one result is returned, we consider that there is ambiguity regarding the search, so the user will have to choose between various concepts to get the one he actually needs, see Fig. 8. Here, in order to let him easily choose between various concept, our system displays the label and type of each class or instance.



Fig. 8: Choosing the right concept for a given tag

Then, a basic SPARQL query, using the `sioc:topic` property, will retrieve all blog posts for the given concept (`<$uri>`), that can be ordered thanks to `sioc:Post` properties as creation date:

```
SELECT ?post ?date ?author
WHERE {
  ?post sioc:topic <$uri> .
  ?post dc:created ?date .
  ?post sioc:has_creator ?author
} ORDER BY ASC(?date)
```

The interface will then present results as a classical search engine, with links to original blog posts where users can see that tags are different from one post to another, but express the same idea, i.e. the related ontology class or instance.

4. Offering a better search experience to users

Since one of our goal is to enhance information retrieval capabilities within our platform, we introduced different ways to extend search by offering suggestions of related concepts (and so related posts) to our users. Thus, we defined various methods that find related posts using different algorithms or points of view, that wil be defined in the next sections.

Moreover, as each method returns a set of ontology classes or instances, associated with blog posts, we decided to provide a different tag cloud for each set of suggestions, see Fig. 9.

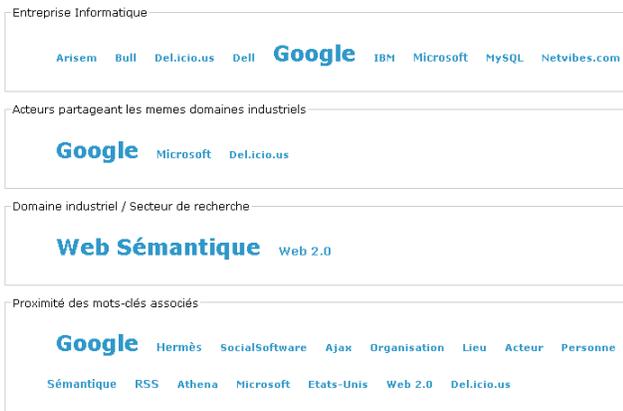


Fig. 9: Tagclouds of related concepts

4.1 Statistical approach

A first way to discover related blog posts from a given one is to use the co-occurrence between concepts that are available on the platform.

Thus, for each post dealing with concept A, the system will suggest the user to look as posts linked to concept B as soon as the concepts appear in the same blog post, i.e.:

```
:p rdf:type sioc:post .
:p sioc:topic :A .
:p sioc:topic :B
```

In the future, we can imagine to add a threshold property to each class to consider as related only classes or instances that has a minimal number of co-occurrences, which can vary regarding their type.

4.2 Using ontology and instance type

Another approach to infer related concepts and associated posts is to use the ontology and especially classes and their instances.

Thus, if dealing with a post linked (using `sioc:topic`) to an instance of `ProgrammingLanguage` (eg: PHP), all other posts linked to a `ProgrammingLanguage` instance will be considered as related, and the interface will suggest the user to look at them when searching for PHP.

From a theoretical point of view, we are using the following rule:

```
{
  :x rdf:type sioc:post .
  :y rdf:type sioc:post .
  :x sioc:topic :a .
  :y sioc:topic :b .
  :a rdf:type :t .
  :b rdf:type :t
} => {
  :x sioc:related_to :y
} .
```

Once again, SIOC offers some useful properties that can be used in this context, since we consider two posts dealing with instances of the same class as semantically related thanks to the `sioc:related_to` property, see Fig. 10.

Actually, in practice, we did not implement this rule directly in our RDF store, even if it was technically possible. Indeed, as mentioned before, we want to make differences between related posts depending the way they have been defined as related. By inferring this rule, there will not bt any difference regarding the way the post have been considered as related, as there is a single `sioc:related_to` relationship. We could have create subproperties, yet we decided to trigger the rule when the user search for blog posts.

Thus, we implemented the following SPARQL query that is called when a user is looking for a given concept and that returns posts that would have been set as related using the previous rule, and associated topics so that we can create the needed tagcloud:

```
SELECT ?post ?topic
WHERE {
  <$uri> rdf:type ?type .
  ?post sioc:topic ?topic .
  ?topic rdf:type ?type
}
```

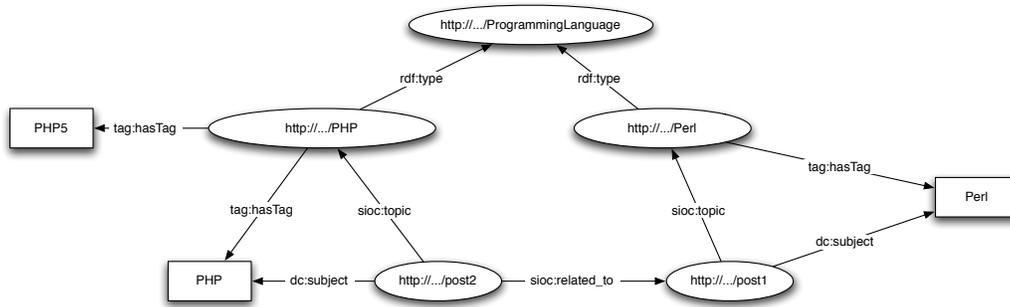


Fig. 10: Linking related posts with *sioc:related_to*

4.3 Defining specific rules

Going further, and in order not to limit related posts only to the ones that use instances of the same classes, we defined inference rules than can be independently associated to ontology classes, in order to specify what we should consider as a related class for a given one.

For example, for the `Location` class, we defined the following pattern:

```
{
  :x rdf:type sioc:post .
  :y rdf:type sioc:post .
  :x sioc:topic :a .
  :y sioc:topic :b .
  :a pup:subLocationOf :b
} => {
  :x sioc:related_to :y
} .
```

This rule means that the system will suggest related sublocations when looking for any tag associated to an instance of `Location` or its subclasses. Thus, our interface will suggest the user to look at posts linked (using `sioc:topic`) to `Paris` - as the capital of France, not Paris Hilton, avoiding eventual ambiguity - when looking for items tagged with `France`.

Another interesting rule is the following one, which will suggest all `Agent` - and its subclasses - instances that are experts in the same domain, as shown in Fig. 9:

```
{
  :x rdf:type sioc:post .
  :y rdf:type sioc:post .
  :x sioc:topic :a .
  :y sioc:topic :b .
  :a rdf:type ptop:Agent .
  :b rdf:type ptop:Agent .
  :a company:expertiseIn :d .
  :b company:expertiseIn :d
} => {
  :x sioc:related_to :y
} .
```

Once again, we implemented these rules using a live SPARQL query, in order to create different clusters regarding the rules:

```
SELECT ?post ?topic ?domain
WHERE {
  <${uri}> [
    rdf:type ptop:Agent ;
    company:expertiseIn ?domain
  ] .
  ?post sioc:topic ?topic .
  ?topic [
    rdf:type ptop:Agent ;
    company:expertiseIn ?domain
  ]
}
```

In order to manage these inference rules from a semantic web point of view, we introduced a `RelatedTag` class in our ontology, which has the following properties, and that can be associated to any `rdf:Resource`:

- a label that will be shown in the user interface so that the user can know why it should look at the following posts;
- the associated rule. Even if we will not use it right now, it is useful to store it in case in the future we decide to implement it directly on our RDF store for any particular reason (for example: some dedicated stores to communities of interest);
- the corresponding SPARQL pattern, which is used in our implementation.

5. Conclusion and future works

In this paper, we introduced an approach that mixes folksonomies and ontologies, in order to get rid of various issues regarding free-tagging classification. Thanks to add-ons to an existing blogging platform, and by implementing a semantic search engine, we showed how this method could enrich information retrieval, by solving ambiguity and tags explosion, and also suggesting related posts in various ways.

Moreover, we also introduced how this use lead to social ontology population, as the blogging community can help to create new instances of the ontology.

One aspect of our future works will consist in establishing communities of interest thanks to this architecture. Indeed, as we can now link posts to ontology concepts, we might be able to infer connections between not only people that are using common tags - which can be done with a single folksonomy - but with people blogging about similar concepts

(i.e. people blogging about instances of the same class as `ProgrammingLanguage`, or classes that have strong relationships). Thus, we hope to see new communities of interest appear, and maybe that could not have been guessed using a basic free-tagging scheme.

Another part of our work should also consist in information extraction from blog posts, once again using ontologies and available properties between instances that have been associated to blog posts.

Acknowledgements

We would like to thank all the members of EDF R&D that have been involved in the project, especially bloggers and platform administrators (id-net group), for their participation as they helped to gather information sharing and weblogging through the company.

References

- [1] B. Adida and M. Birbeck. RDFa Primer 1.0: Embedding RDF in XHTML. <http://www.w3.org/TR/xhtml-rdfa-primer/>, May 2006. W3C Working Draft.
- [2] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 284(5):35–43, 2001.
- [3] J. G. Breslin, A. Harth, U. Bojars, and S. Decker. Towards Semantically-Interlinked Online Communities. In *The 2nd European Semantic Web Conference (ESWC '05), Heraklion, Greece, Proceedings, LNCS 3532*, pages 500–514, May 2005.
- [4] D. Brickley and R. Guha. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation. <http://www.w3.org/TR/rdf-schema/>.
- [5] D. Brickley and L. Miller. FOAF Vocabulary Specification. <http://xmlns.com/foaf/0.1/>.
- [6] S. Cayzer. What next for Semantic Blogging? Technical Report HPL-2006-149, Hewlett-Packard Laboratories, Bristol, UK, Oct. 2006.
- [7] S. Cayzer and P. Castagna. How to build a Snippet Manager. In S. Decker, J. Park, D. Quan, and L. Saueremann, editors, *Proc. of Semantic Desktop Workshop at the ISWC, Galway, Ireland, November 6*, volume 175, November 2005.
- [8] D. Connolly. Gleaning Resource Descriptions from Dialects of Languages (GRDDL). <http://www.w3.org/2004/01/rdxh/spec>.
- [9] DCMI Usage Board. The Dublin Core Metadata Initiative. <http://dublincore.org/documents/dcmi-terms/>.
- [10] W. G. Dowling. Corporate weblogs: Deployment, promotion, and measurement. Technical report, Jupiter Research, June 2006.
- [11] S. Golder and B. A. Huberman. The Structure of Collaborative Tagging Systems. *Journal of Information Science*, 32(2):198–208, April 2006.
- [12] T. R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, June 1993.
- [13] T. R. Gruber. Ontology of Folksonomy: A Mash-up of Apples and Oranges. Invited keynote, November 2005.
- [14] S. Harris. SPARQL query processing with conventional relational database systems. In *International Workshop on Scalable Semantic Web Knowledge Base System (SSWS 2005)*, 2005.
- [15] F. Manola and E. Miller. RDF Primer. <http://www.w3.org/TR/rdf-primer/>, February 2004.
- [16] A. P. McAfee. Enterprise 2.0: The Dawn of Emergent Collaboration. *MIT Sloan Management Review*, 47(3):21–28, 2006.
- [17] K. Möller, J. G. Breslin, and S. Decker. semiBlog - Semantic Publishing of Desktop Data. In *14th Conference on Information Systems Development (ISD2005), Proceedings, Karlstad, Sweden, August 2005*.
- [18] R. Newman. Tag ontology design. <http://www.holygoat.co.uk/projects/tags/>.
- [19] A. Passant, P. Laublet, and J. D. Sta. Folksonomies, Ontologies and Corporate Blogging. In *BlogTalk Reloaded*, Vienna, Austria, October 2006.
- [20] E. Prud'hommeaux and A. Seaborne. SPARQL Query Language for RDF. W3C Working Draft, W3C, 2006. <http://www.w3.org/TR/rdf-sparql-query/>.
- [21] I. Terziev, A. Kiryakov, and D. Manov. Base Upper-level Ontology (BULO) Guidance. Technical report, July 2005. Deliverable 1.8.1, SEKT project.
- [22] T. V. Wal. Folksonomy. <http://vanderwal.net/folksonomy.html>.