Schema Based Annotation Method for Information Management in Semantic Web

Win Win Naing, Naychi Lai Lai Thein
University of Computer Studies, Mawlamyine
naingnaing.ww.89@gmail.com, naychillt@gmail.com

Abstract

Currently, much of the information on the Web is described using only natural language, which can be seen as a major obstacle in developing the Semantic Web. Since the annotations describing different resources are one of the key components of the Semantic Web, easy to use and cost-effective ways to create them are needed, and various systems for creating annotations have been developed. However, there seems to be a lack of systems that can be easily used by annotators unfamiliar with the technical side of the Semantic Web, and are able to support distributed creation of semantic metadata based on complex metadata annotation schemas and domain ontologies.

In this paper, we have developed an annotation system with concept ontology. Then, the domain ontology is created from the annotators. Our system adapts flexibly to different metadata schemas, which makes it suitable for different applications but the system need to predefined synonyms terms.

1. Introduction

The advancement of various data and information management technologies contributed to the rapid growth of the World Wide Web. One of the major problems with the Internet is information overload. Because humans can now access large amounts of information very rapidly, they can quickly become overloaded with information and, in some cases, the information may not be useful to them. In certain other cases, the information may even be harmful to the humans. The current search engines, although improving steadily, still give the users too much information. When a user types in an index word, many irrelevant web pages are also retrieved. Because the Web pages currently are for human consumption and manipulation. So, we need the Web pages to be understood by machines. This is the idea behind the Semantic Web. Semantic Web will extend the current Web with representation of semantics to read and understand the contents of Web pages [4].

The Semantic Web, XML, and Semi structured database are still relating new technologies and include many other technologies. Some technologies for Semantic Web include Resource Description Framework (RDF), ontology, agents and database. Ontologies are vital for developing the Semantic Web. XML and RDF are special way of representing the various ontologies [2].

The process of document annotation for the Semantic Web is complex and time consuming as it requires a great deal of manual annotation. Moreover, users generally can't know the terms used in the database. If user enters the keyword that have same meaning but have different symbols with the prestored words, the system cannot produce the output result for the user's request. Because of the machine is not understood. This is also called terms synonyms. This system can handle the terms synonyms by using terms ontology. So, the system becomes machine-understandable and easy to use for the users.

In this paper, related work and problem issues are discussed in section 2. Section 3 presents the proposed system architecture and section 4 describes the framework for the schema based annotation system. Conclusions are then discussed in section 5.

2. Related Work and Problems Issues

The Semantic Web is an extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in cooperation. The realization of the Semantic Web requires the wide-spread availability of semantic annotation for existing and new document on the Web. Semantic annotation is the process of interesting tags in the document, whose purpose of inserting tags is to assign semantics to the text between the opening and closing tags. The content of the Web documents is annotated inside the document tags. Thus, information is not repeated and redundancy is avoided.

In Internet, there are vast amounts of free text that are neither grammatical nor formally structured. These sources of data, called "posts," are full of useful information, but they lack the semantic annotation to make them searchable. Annotating these posts is difficult since the text generally

exhibits little formal grammar and the structure of the posts varies. However, by leveraging collections of known entities and their common attributes, called "reference sets". To use this reference data, we align a post to a member of the reference set, and then exploit this matched member during information extraction [3].

To revolutionize the use of the Internet, we have to face some major challenges. First, construction of the Semantic Web requires a lot of extra markup on documents. Second, there is a lot of information that would be more useful if it were annotated for the Semantic Web, but the nature of the data makes it difficult to do so. It would be beneficial to add semantic annotation to each piece of text from the input statement ("post") as shown in Figure 1. The annotation tasks carry no burden to human users [3].

Example input statement:

Who is the **professor** of the **Maryland Univ**.?

<*occupation*>**professor** <*/occupation*>
<*Ref_occupation*> **Professor** <*/Ref_occupation*>
<*universityName*> ***M*aryland Univ.**
<*universityName*>
<*Ref_universityName*> **MarylandUniversity**
<*/Ref_universityName*>

Figure 1. A post from the Input Statement

The process of document annotation for the Semantic Web is complex and time consuming as it requires a great deal of manual annotation. This paper proposes the idea of schema based annotation system for terms ontology and describes the development of information ontology. This system can handle the terms synonyms by using the concept ontology. So, the systerm becomes machine-understandable and easy to use for the users.

3. Schema Annotated Information Management

Ontology-based semantic annotations are needed when building the Semantic Web. Although various annotation systems and methods have been developed, the question of how to easily and cost-effectively produce quality metadata still remains largely unanswered. We tackled the problem by first identifying the major requirements for an annotation system.

As a practical solution, annotation systems is designed and implemented which supports distributed creation of metadata and utilize ontology services as well as automatic information extraction. It is designed to be easily used by non-experts in the field of the Semantic Web. Currently, the coupling of the annotation schema's properties and information extraction components are not fully utilizing the ontological characteristics.

To accomplish the semantic annotations, we defined a set of domain and annotation ontologies. We then selected a representative set of heterogeneous cultural contents of different kinds and annotated them with metadata conforming to the designed ontologies and annotation schemas. The result is homogenized based on the action of concept ontology. After this, the view-based semantic search engine and logical recommender system is adapted and applied to the new content set.

Continuing with our example from Figure 1, assume there is ontology of terms synonyms, and from it we build a reference set with the following attributes: occupation, universityName etc. To use reference sets for semantic annotation we exploit the reference set to determine which, if any, of the attributes appear in the post. To do this, we first determine which member of the reference set best matches the post. We call this the record linkage step. Then we exploit the attributes of this reference set member for the information extraction step by identifying and labeling attributes from the post that match those from the matching member of the reference set. We annotate the post in this manner [3].

For instance, the post for the input statement in Figure 1 matches the reference set member with the occupation of "Professor" and the universityName of "Maryland University." Using this match we label the tokens "maryland univ." of the post as the "universityName," since they match the universityName attribute of the matching reference set record**.** In this manner we annotate all of the attributes in the post that match those of the reference set. For purposes of exposition, the reference set shown only has two attributes: occupation and universityName [3].

In addition to annotating attributes in the post from the reference set, we also annotate attributes that are identifiable, but not easily represented in reference sets. Also, we include annotation for the attributes of the matching reference member. (These are called "Ref occupation..." in Figure 1). Since attribute values differ across posts, these reference member attributes provide a set of normalized values for querying. Also, the reference set attributes provide a simple visual validation to the user that the IE step identified things correctly. Lastly, by including attributes from the matching reference member, we can provide values for attributes that were not included in the post [3].

3.1. Defining the Concept Ontology

The concept ontology is the ontology that defines how things in the world relate to each other. The actions, the objects, the places, indeed everything that is present in situations are defined in a large ontology that tells what the semantic connections between the concepts are.

As the processes are made up of situations, the situations are made up of different properties of the situation. All these properties have a value that is a reference to a resource of the content definition ontology. For example, consider the situation where the trees are cut down in the slash and burn method. In the action-property of the situation, there will be a reference to the concept of cutting down, and the object of action is a reference to the concept of tree.

With all the concepts and their relations are defined, create links between different processes that have to do with trees, even though in another process or situation, the annotation would have been oaks. The ontology tells the computer that oaks are a subclass of trees.

3.2. Schema Annotation

Today, Standard Information Extraction Technology has been adapted for retrospective archive search. Since the system relies on high quality human annotated training data for constructing named entity recognizers (NERs), any inconsistency introduced into the annotation schema by ontological inconsistencies should be harmful for annotation performance, both human and machine. In the semantic web, the annotation schema is a dependency-based format augmented with null elements and enriched through a refinement of grammatical relations [1].

In this paper we present to annotate the database schema and base the annotations directly on the schema. In the schema based annotation, things to be extracted are defined by the properties of the annotation schema's classes. For example <occupation name=" faculty" type="string"> **Professor**</occupation>, means that the entity mentioned by "**Professor**" is related to the class faculty and occupation is the property of the faculty class. The reason for using this expression is to cover two relations between mentioned entities and the ontology we want to describe [1].

The first is "is an instance of '', and the other one is "is a subclass of''. Some of the markable texts mention a particular and others mention a universal. For example, names of persons, occupation, locations and organizations are usually used to refer to a particular, whereas names of chemical substance, viruses and proteins are often used to refer to universals [1, 4].

Accordingly, the function of an extraction component is to provide suitable concepts or entities to be used as values of those properties. Because it supports arbitrary annotation schemas, extraction method must be adaptable in order to support different extraction tasks.

Implementation steps are described as follow:

Step 1: Input statement.
Step 2: Extract and convert the keywords from the statement by using annotation.
Step 3: Match the extracted keyword with the prestored terms ontology.
Step 4: Extract the desired information from the information ontology by using schema based annotation.
Step 5: Output the result.

A typical example of proposed system implementation is shown in figure 2. In ontology, all categories of concepts are represented as classes which follow a disjoint entity class principal that has been the underlying premise of NERs. The corresponding annotation schema will also be simpler, since instances of context-dependent classes are annotated in the same way as those of other classes, for example:

<occupation name= "faculty" type= "string">
Professor
</occupation>
<universityName name = "university" type= "string">
Maryland University
</ universityName>

3.3 Extracting Keywords by using XML-Query

XQuery is a query language for real and virtual XML documents and collections of these documents. In this system, after annotating the input statement we extract the keyword to match with the prestored terms synonyms ontology by using XQuery. It was devised primarily as a query language for data stored in XML form. So its main role is to get information out of XML databases — this includes relational databases that store XML data, or that present an XML view of the data they hold. Some people are also using XQuery for manipulating free-standing XML documents, for example, for transforming messages passing between applications [7, 6].

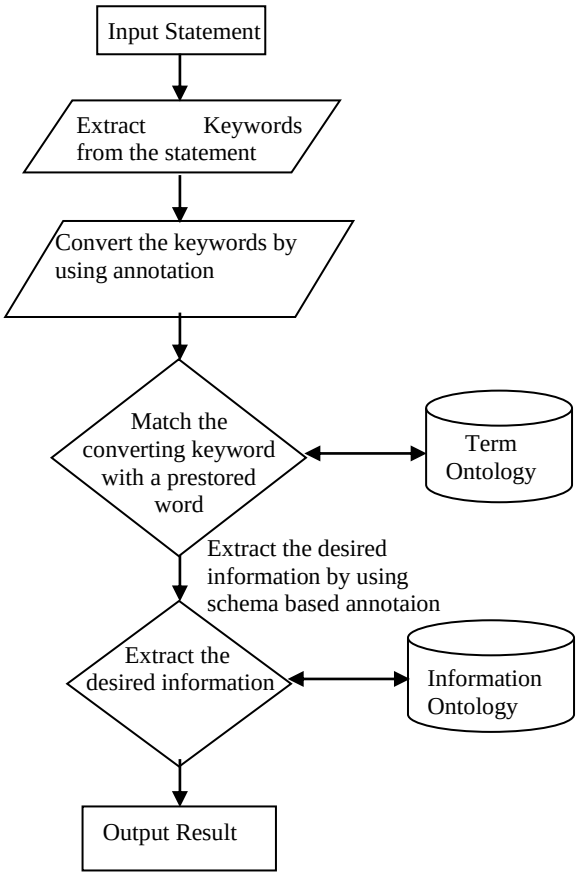

Figure 2. Design of the system

XQuery language allows selecting elements/attributes from input documents. It can join data from multiple documents and made modifications to the data. Calculate new data and add new elements/attributes to the results. Moreover, it is used for extracting information from the database for use in a Web service and for searching textual document on the Web for relevant information and compiling the results. For example:

Input Document: Resulted XML document.

<occupation name= “faculty” type= “string”>
Professor
</occupation>
<universityName name = “university” type= “string”>
Maryland University
</ universityName>

XML_Query:
String (input () //occupation [name = "faculty"], universityname)

Output:
‘Professor’, 'Maryland University'

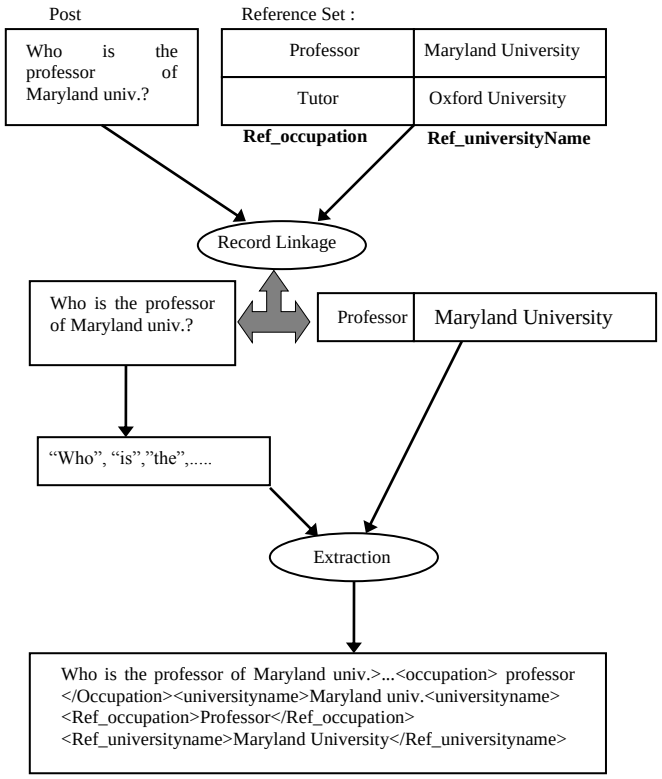

Figure 3. Annotation Algorithm

4. Solving Terms Synonyms in Ontology

Generally, users can’t know the terms used in the database because of there are many database in the Semantic web. Although there is user’s request in the database when the user enters the keyword that have same meaning but have different words with the prestored words, user’s request can’t output. Because of the machine is not understood.

This system can handle the terms synonyms. Firstly, prestored ontology is developed to solve terms synonyms, and then search.

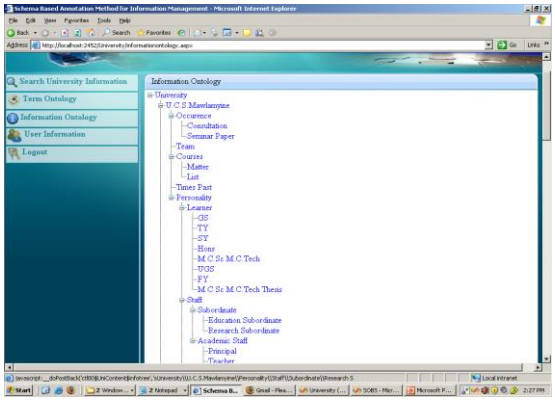

Figure 4. Information Ontology

In solving terms synonyms, the system is search the desired keywords in the terms ontology. When the resulted keywords are found, the system is output the root node of the search keywords. In the above example, the system is output '**Rector**' instead of '**Professor**'. After solving the terms synonyms, the resulted keyword is used to search the desired information. In this step, the XQuery is also used to search the user request and then the output. For example:

Input Keywords: "Rector", "Maryland University"

XML_Query: University/ University [@ universityname = "Maryland University"]/ Staff/Staff [@AdminStaff="Rector"]

Output: "Dr.Smith"

However, the system used terms ontology to solve terms synonyms. Because of the Maryland University's ontology saved **Rector** instead of **Professor.** Terms synonyms can understand by human visions but machine can't understand. So, by using terms ontology as a medium the system can be machine-understandable. Figure 4 describes the information ontology.

5. Implementation Results

This section depicts the implementation results of the Schema Based Annotation Method for Information Management System.

When the user is entered valid User Name, Password, *Search page* will appear which is shown in figure 5. If the user is the administrator, he/she can manage the user and the role. If the user is not administrator, he/she can only search the required data. The user can not view the terms ontology and the information ontology.

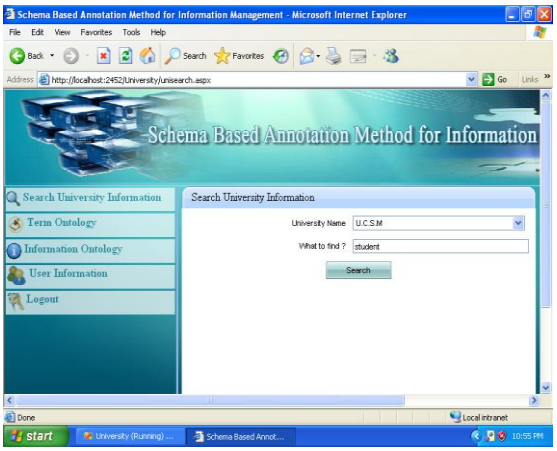

Figure 5. Inquiry from the Search Page

In the Search page, *Inquiry* is intended for searching user desired data. If the user desired data has in the ontology database, the result is shown as in figure 6.

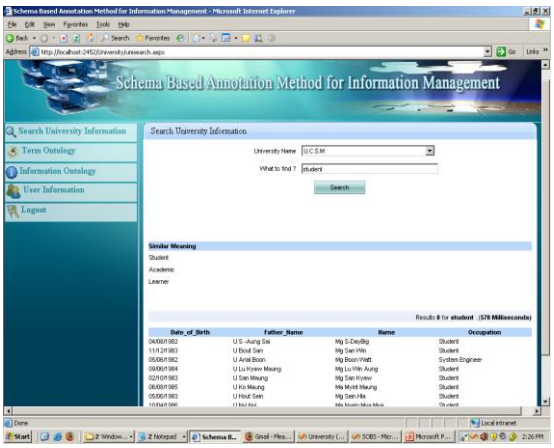

Figure 6. The resulted search page

Figure 7 show the predefined synonyms terms. The system searches the user's requested data according to the synonyms terms such as Head, Principal, Professor, Chief, Rector and Chairman. The user requested data is "the name of the head of Maryland University", if the usage of Maryland University is Principal, the system will give the name result of Principal of the Maryland University.

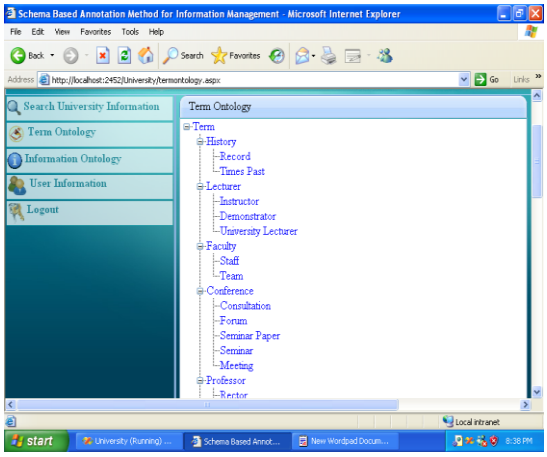

Figure 7. Terms Synonyms

6. Conclusions and Further Extensions

In this paper, we present development of concept ontology and information ontology. Concept ontology was created to provide the common annotation scheme to describe situations, their main properties and define the concepts appearing in the situations unambiguously. It relates a human-understandable domain description with a machine-understandable description. Moreover, schema based annotation approach is described in solving terms conflict among multiple heterogeneous ontologies. By solving terms conflict, the Web will be machine-understandable. So, it will allow the development of high quality techniques for automated discovery on the Web, stepping toward seamless information searching of applications and data on the Web.

The advantage of the system is the semantic knowledge of the web page and there is a direct association defined between the ontology and the database implementation. For example, changing a web page without altering the meaning of the content, the system does not need to modify the annotations. The annotation has to be defined only once and the same concept can be reused in different object.

The system cannot provide the semantic autocompletion and linguistic concept extraction.

Our future plan is to provide metadata for additional semantic portals as well as further develop the automation of the annotation and the linguistic concept extraction.

References

[1]. A. Kawazoe, L. Jin, M. Shigematsu, R. Barrero, K. Taniguchi, N. Collier, "The development of a schema for the annotation of terms in the BioCaster disease detecting/tracking system".

[2]. B. Thuraisingham, "XML Databases and the Semantic Web".

[3]. M. Michelson and C. A, "Semantic annotation of unstructured and ungrammatical text", Knoblock University of Southern California Information Sciences Institute, 4676 Admiralty Way Marina del Rey, CA 90292 USA {michelso,knoblock}@isi.edu

[4]. N. F. Noy and D. L. McGuinness, ". Ontology Development 101: A Guide to Creating Your First Ontology", Stanford University, Standardford, CA, 94305.

[5]. N. Memon, O. I. Eldai and M. A. Uniquali, "Semantic Web Languages: A Comparison", Processing of the Third International Conference on Computer Application, Yangon Myanmar, Page 385.

[6]. **XQuery 1.0 and Web Applications** Mary Fernández AT&T Labs – Research Jêróme Siméon IBM Watson Research Center

[7]. XQuery Tutorial Peter Fankhauser, Fraunhofer IPSI Peter.Fankhauser@ipsi.fhg.dePhilip Wadler, Avaya Labswadler@avaya.com