Semantic Information Retrieval Using WAMP Server

Mihir Chakrabarty
Department Of Information Technology, Dronacharya College of Engineering, Haryana 122 506, India

***Abstract*- With the rapids growth of internet and its global use, users find it difficult to get meaningful information from it. To overcome this problem, semantics methods for retrieval of required data are being studied. Information retrieval technology has been central to the success of the Web. For semantic web documents or annotations to have an impact, they will have to be compatible with Web based indexing and retrieval technology. This research paper explores the possibility of extracting semantic based retrieval of information. First ontology was built which was published on the web. Then the ontology was loaded to MySQL data store. Using PHP language information from this ontology is retrieved using SPARQL. WAMP server which is a package comprising of Apache, MySQL and PHP is used for the process. This paper demonstrates the dominant nature of information retrieval from the web using semantic technologies.**

***Index Terms*- ARC, Knowledge base, semantic web, SPARQL, WAMP server.**

I. INTRODUCTION

World Wide Web is growing at an alarming rate. It is composed of documents written in Hypertext Markup Language (HTML) which is a huge collection of unstructured data. Information retrieval has been central to the Web. Everyday thousands of pages are added. Managing this large amount of data is a difficult task. Information in the current web is only for human consumption. This has made extracting information difficult. Tim Berners Lee, founder of World Wide Web, recognized its big potential and coined semantic web, his vision for the next generation of the web. The Semantic Web [1] allows the representation and exchange of information in a useful way, facilitating automated processing of descriptions on the Web.

One vision of the Semantic Web is that it will be much like the Web we know today, except that documents will be enriched by annotations in machine understandable markup. Semantic web enriches human readable data with machine readable annotations. It will be simple for machines to understand such a web. Annotations explicitly express links between different web resources and connect these resources to formal terminologies. Such structures are called ontologies. Ontologies provide common vocabularies to be used on the Semantic Web. To further simplify the data integration and automation work, W3C has developed meta data standards such as Resource description framework (RDF) [2], and the web ontology language (OWL).

II. MOTIVATION

The Semantic Web has lived its infancy as a clearly delineated body of Web documents. That is, by and large researchers working on aspects of the Semantic Web knew where the appropriate ontologies resided and tracked them using explicit URLs. When the desired Semantic Web document was not at hand, one was more likely to use a telephone to find it than a search engine. This closed world assumption was natural when a handful of researchers were developing DAML 0.5 ontologies, but is untenable if the Semantic Web is to live up to its name. Yet simple support for search over Semantic Web documents, while valuable, represents only a small piece of the benefits that will accrue if search and inference are considered together. We believe that Semantic Web inference can improve traditional text search, and that text search can be used to facilitate or augment Semantic Web inference. Several difficulties, listed below, stand in the way of this vision. Current Web search techniques are not directly suited to indexing and retrieval of semantic markup. Most search engines use words or word variants as indexing terms. When a document written using some flavor of SGML is indexed, the markup is simply ignored by many search engines. Because the Semantic Web is expressed entirely as markup, it is

thus invisible to them. Even when search engines detect and index embedded markup, they do not process the markup in a way that allows the markup to be used during the search, or even in a way that can distinguish between markup and other text. Current Web search techniques cannot use semantic markup to improve text retrieval. Web search engines typically rely on simple term statistics to identify documents that are most relevant to a query.

III. RDF, ONTOLOGY AND SPARQL

RDF, Ontology and SPARQL form the three core component of semantic web. Semantic web is a web of databases and not of documents, queried by SPARQL [7]. Ontology, RDF and SPARQL collectively play important role in transforming current web to semantic web.

A. RDF

RDF [4] is the first language developed especially for the Semantic Web. It is recommended by W3C for writing machine processable annotations. RDF defines resources using XML. RDF is also called triple because it has three parts subject, object and predicate. Subject and object are names for resources and predicate is the relationship that connects these two things.

All Information can be represented in the form of triples. RDF represents relationship between any two data elements, allowing for a very simple model. Below shows information about Carrot expressed as subject predicate object.

Subject	predicate	object
Carrot	is_a	Vegetable
Carrot	HasAlphaCarotene	3477
Carrot	BetaCarotene	8509
Carrot	LuteinZeaxanthin	256
Carrot	HasLycopene	1
Carrot	HasPlantType	Biennial
Carrot	HasVegType	Root

Table 1: Part of the RDF triple relationship for Carrot

B. Ontology

Ontologies link computer and human understanding of symbols. These symbols are also called as relations. Ontology is a specification of a shared conceptualization [3]. Ontology is specific to a domain, and it represent *an area of* knowledge Hence users and domain experts should agree on the knowledge being represented by ontology so that it can be shared and reused.

C. SPARQL

SPARQL was standardized by W3C. SPARQL is a query language that is used to query RDF data. It can also be used to query remote RDF server. Like RDF, basic building block of SPARQL query is the triple pattern. A triple pattern is like a triple, but it can have variables in place any of the three positions: subject, predicate or object.

IV. DESIGN METHODOLOGY

The key steps for semantic information retrieval using WAMP server is as follows

Workflow steps:

1. Install WAMP server
2. Install ARC
3. Configure MySQL
4. Configure SPARQL Endpoint
5. Create Ontology
6. Load ontology into Mysql
7. Query using SPARQL in PHP

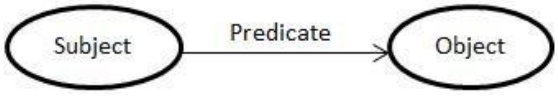

Fig1: Structure for RDF

V. IMPLEMENTATION

This section describes the implementation of the proposed approach for semantic information retrieval.

A. Install WAMP Server:

For this purpose, open source WAMP server was downloaded and installed. It groups programs that are run on Microsoft windows operating system.

WAMP is an acronym formed from the initials of the operating system Microsoft Windows and the principal components of the package: Apache, MySQL and one of PHP, Perl or Python. Apache is a web server. MySQL is an open-source database. PHP is a scripting language that can manipulate information held in a database and generate web pages dynamically each time content is requested by a browser.

B. Install ARC:

ARC provides a RDF system for PHP developers of semantic web. It is an open source, easy to use system. It runs in almost all of the web server environments.

C. Configure MYSQL server:

In local MYSQL server, create the database. Create a user and give all permissions on this database to the user.
To create RDF store in MYSQL, augment config.php file with the following code:-

```
$store = ARC2::getStore($config);
if (!$store->isSetUp()) { $store->
->setUp();}
```

D. Configure SPARQL Endpoint:

A *SPARQL endpoint is* an interface that users use to query an RDF data store by using SPARQL query language. It is a machine friendly interface towards knowledge base. It accepts queries and return result accordingly. This endpoint could be a stand-alone or a web based application. Endpoint returns results in a number of different formats like HTML table, RDF/XML, Turtle, JSON.

Most of the time, results are returned in the form of a HTML table, which is constructed by applying XSL transformations to XML result.

SPARQL endpoints can be classified as generic endpoints and *specific* endpoints. If the endpoint is tied to a specific dataset, it is called as specific endpoint. If the endpoint can query any RDF dataset that is stored locally or accessible from the web, it is called as generic.

SPARQL endpoint can be configured by giving appropriate host name, database name, database user name, database password and store name.
This paper uses ARC SPARQL endpoint to query the RDF datastore.

E. Create Ontology:

Example ontology is created for nutrition domain for Vitamin A. It has DietarySource, Disease, EffectsOnHuman, FormsofVitaminA, Interaction, Person, Supplement as the main classes. Ontology is created as class subclass relationship.

Ontology also defines properties and restrictions on the domain. Individuals also called instances are created. Ontology is then published on the web. For each instances data properties are defined like AlphaCarotene, BetaCarotene, LuteinZeaxanthin, Lycopene, HasName,HasPlantType, HasQuantity, HasUnitOfMeasurement, HasVegType. Once ontology is finalised, it is published on the web.

F. Load data into MySQL:

After finalising the ontology, it has to be loaded into RDF data store, MySQL. There are two ways to load OW/RDF data into MYSQL OWL/RDF data can be loaded into MYSQL either through command line or through PHP application code. Whichever is the method used to load OWL/RDF data into MYSQL, the basic load statement is given below. Either this can be executed at the command prompt or inserted in a PHP application code.

```
$store->query(„LOAD
<http://mu123.site90.net/vit2.owl>');
```

G. Retrieval of Information:

Once data is loaded into RDF data store, SPARQL query can be embedded into PHP code to get the desired result.

Below shows PHP code using SPARQL to retrieve information regarding the resource name and their vitamin A content, from the RDF store.

```
<?php
include_once("arc/ARC2.php");
$store =
ARC2::getStore($arc_config); if
(!$store->isSetUp())
{ $store->setUp();
} $q='PREFIX ta:
<http://www.owl-
ontologies.com/vit2.owl#> SELECT *
from
http://mu123.site90.net/vit2.owl
WHERE
{

        ?s1 ta:HasName ?s.

        ?s1 ta:HasVitaminAinIU ?p.
}';

$rows = $store->query($q,
'rows'); $r = '';
if ($rows = $store->query($q, 'rows'))
{
$r = '<table border=1>
<th>Name</th><th>VitaminA</th>'."\n";
$r .='</table>'."\n";
}

Else
{ $r = '<em>No data returned</em>'; } echo
$r;
?>
```

The output of the given code is shown below

Name	VitaminA
Apricot	1926
Papaya	1094
Spinach	9376
Beef	0
Mango	765
Parsley	8425
ChestNut	28
Melon	569

VI. THREE PROTOTYPE SYSTEMS

We have explored the problems and approaches to solving them through three prototype systems. While these systems do not exhaust the space of possibilities, they have challenged us to refine the techniques and provided valuable experience. The first prototype, OWLIR, is an example of a system that takes ordinary text documents as input, annotates them with semantic web markup, swangles the results and indexes them in a custom information retrieval system. OWLIR can then be queried via a custom query interface that accepts free text as well as structured attributes. Swangler, our second prototype, is a system that annotates RDF documents encoded in XML with additional RDF statements attaching swangle terms that are indexi- ble by Google and other standard Internet search engines. These documents, when available on the web, are discovered and indexed by search engines and can be retrieved using queries containing text, bits of XML and swangle terms. Our third prototype is Swoogle, a crawler-based indexing and retrieval system for RDF documents. It discovers RDF documents and adds metadata about them to its database. It also inserts them into a special version of the HAIRCUT information retrieval engine that uses character n-grams as indexing terms.

OWLIR is an implemented system for retrieval of documents that contain both free text and semantic markup in RDF, DAML+OIL or OWL. OWLIR was designed to work with almost any local information retrieval system and has been demonstrated working

with two–HAIRCUT and WONDIR. Currently the semantic web, in the form of RDF and OWL documents, is essentially a web universe parallel to the web of HTML documents. There is as yet no standard way for HTML (even XHTML) documents to embed RDF and OWL markup or to reference them in a standard way that carries meaning. Semantic web documents reference one another as well as HTML documents in meaningful ways. Some Internet search engines, such as Google, do in fact discover and index RDF documents. There are several problems with the current situation that stem from the fact that systems like Google treat semantic web documents (SWDs) as simple text files. One simple problem is that the XML namespace mechanism is opaque to these engines. A second problem is that the tokenization rules are designed for natural languages and do not always work well with XML documents. Finally, we would like to take advantage of the semantic nature of the markup.

Since the current semantic web consists of documents encoded in RDF, it is worth considering what a specialized indexing and retrieval engine for these semantic web documents (SWDs) might be like. Search engines for SWDs could exploit the fact that the documents they encounter are designed for machine processing and understanding. Conventional search engines can not do much to interpret the meaning of documents because the state of the art in natural language processing is not up to the task. Even if it were, the computational cost for interpreting billions of documents would be prohibitive in any foreseeable future. SWDs, on the other hand, are encoded in languages designed for machine interpretation and understanding

VII. CONCLUSION AND FUTURE WORK

As of today Web consists of millions of web pages. Data is represented in HTML pages. Hence it is inefficient for meaningful information extraction. This paper focusses on ways to enhance the search results by using ontology and RDF. WAMP server is used to extract information from RDF store where in SPARQL query is embedded in PHP program. This work can be further enriched to achieve intelligent fuzzy retrieval. The semantic web is better suited for data integration and for knowledge representation. RDF and OWL along with SPARQL play an important role in information retrieval from the semantic web. Finally, there is also a role for specialized search engines that are designed to work over collections of RDF documents.

REFERENCES

[1] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, May 2001.

[2] Davis, Ian, "An Introduction to RDF", http://research.talis.com/2005/rdf-intro/

[3] Dieter Fensel · Holger Lausen · Axel Polleres, Jos de Bruijn · Michael Stollberg · Dumitru, Roman John Domingue Enabling SemanticWeb Services, SPRINGER, 2007

[4] G. Klyne and J. J. Carroll, editors. *Resource Description Framework (RDF): Concepts and Abstract Syntax*. W3C Recommendation, 10 February 2004.

[5] www.wampserver.com/en