SemCrawl: Framework for Crawling Ontology Annotated Web Documents for Intelligent Information Retrieval

Vandana Dhingra and Komal Kumar Bhatia

Abstract. Web is considered as the largest information pool and search engine, a tool for extracting information from web, but due to unorganized structure of the web it is getting difficult to use search engine tool for finding relevant information from the web. Future search engine tools will not be based merely on keyword search, whereas they will be able to interpret the meaning of the web contents to produce relevant results. Design of such tools requires extracting information from the contents which supports logic and inferential capability. This paper discusses the conceptual differences between the traditional web and semantic web, specifying the need for crawling semantic web documents. In this paper a framework is proposed for crawling the ontologies/semantic web documents. The proposed framework is implemented and validated on different collection of web pages. This system has features of extracting heterogeneous documents from the web, filtering the ontology annotated web pages and extracting triples from them which supports better inferential capability.

Keywords: Semantic Web, Ontologies, Crawling, Resource Description Framework (RDF), DARPA Agent Markup Language (DAML), Web Ontology Language (OWL), Uniform Resource Identifier (URI).

Vandana Dhingra
Department of Technology,
University of Pune, Pune
e-mail: vandana_dua_2000@yahoo.com

Komal Kumar Bhatia
YMCA University of Science & Technology,
Faridabad
e-mail: komal_bhatia1@rediffmail.com

R. Buyya and S.M. Thampi (eds.), *Intelligent Distributed Computing*,
Advances in Intelligent Systems and Computing 321, DOI: 10.1007/978-3-319-11227-5_19

1 Introduction

The World Wide Web (WWW) has revolutionized the means of data availability on the internet [6]. With the current structural model of the World Wide Web where anyone can easily publish its own document leads to lot of unstructured information and abundance volume. This is posing difficulties for current web crawlers and search engines to gather relevant information and henceforth it is becoming difficult for users to find information with proper precision and recall.

These difficulties have emerged with a solution to the extension of current web termed as semantic web**.** 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 co-operation [1]. In particular, the semantic web provides a mechanism that is very useful for formatting data in machine readable form, linking individual data properties to globally accessible schemas, matching local references to entities against various kinds of standard names, and providing a range of inferences over that data in scalable ways [12]. The main components that distinguishes Semantic Web are –Ontologies, Languages used to represent ontologies, schemas to represent concepts, adding meaning to document data, triples, URI [5]. It is found that semantic markup within documents leads to greater number of relevant documents as compared with text documents [13].

In present there are different kinds of resources in semantic web – HTML documents embedded with metadata, RDF, OWL,DAML[4], RDF embedded XML documents, all these resources cannot be crawled by the current crawlers because of the meaning based linkage as compared to keyword based linkage in traditional web. Hence there is need of design of system that can crawl all these resources, extract the information from the semantic annotated documents which in turn will help out in providing inferential capability. The above discussed requirement for crawler is designed and implemented in this paper.

The paper is organized as: section 2 discuss about the motivation for developing the crawler framework; section 3 describes the related work in this area; section 4 discusses the proposed crawler and the corresponding algorithms for crawling semantic web documents and ontologies; in section 5 implementation of the crawler framework is presented with the validation of the work done and finally in section 6 conclusion is given.

2 Motivation

Crawling the semantic web is different from crawling the web of HTML documents. A traditional crawler starts with some seed URLs, downloads the corresponding documents, analyzes each document to gather further URLs for crawling and does context specific processing of the retrieved contents, like creating the searchable entries in the database. The last steps are repeated until a

stop criterion is met (e.g. no more URLs to crawl, reached a predefined link depth, or gathered a predefined amount of documents) [14]. But these steps cannot be applied for crawling the web documents specified in languages described for the semantic web.

Table 1 Comparison between the traditional web and semantic web

Parameter	Traditional Web	Semantic Web
Basic concept	It is a collection of documents linked by hyperlinks described in languages with syntax that involves keywords.	It is a collection of documents linked by relations with inferential capability, hence adding a meaning to the links as compared to linkages merely via keywords.
Linkages Structure	Linked using an HTML anchor tag (link) which is a keyword reference to another document generally displayed as underlined text.	Linking among the documents are implemented using the rdfs:seeAlso[17] relationship[11]
Linkage Specification	An HTML hyperlink doesn't specify the actual meaning based linkage between documents.	Documents are written in-Resource Description Framework (RDF), which make it possible to specify how concepts are linked to each other.
Crawling	Operates on HTML documents.	Semantic Web crawler operates on RDF, OWL and other semantic web representation languages [10].

Table 1 specifies the difference between the traditional web and semantic web crawler hence arising the need for design of crawler with different features as compared to the crawlers for crawling the traditional web documents. The functioning of semantic crawler is differentiated from normal crawler that normal crawler must only contend with extracting text from possibly invalid, content and subsequent link extraction whereas a semantic web crawler must carry out additional processing task like merging of information resources via inverse-functional- properties; tracking provenance of data; harvesting schemas and ontologies in addition to source data [16]. The above tabular representation clearly specifies that there is a difference between the linkage structures and linkage specification between the two web structures. Because of these different structures regular crawlers are not sufficient to crawl semantic web and special semantic web crawlers should be developed [15]. Hence there is requirement for crawling framework with different specialties to harvest the semantic web and create knowledge base.

3 Related Work

Ontotext RDF crawler [16] downloads interconnected fragments of RDF from the World Wide Web and builds a knowledge base from this data. At every phase of RDF Crawling, a list of URIs to be retrieved as well as URI filtering conditions (e.g. depth, URI syntax) are maintained which is done to download the resources containing RDF iteratively. To enable embedding in other tools, RDF Crawler provides a high-level programmable interface (Java API).

"Slug" a web crawler (or "Scutter") [7] is designed for harvesting semantic web content. Implemented in java using the Jena API, it works like web crawler, but it fetches RDF files instead of HTML pages, and follows rdfs: seeAlso links instead of HTML links. It has been designed as command based crawler system and does not provide any methods to reuse the crawled data and hence limiting the scope to just crawling of few documents and not providing the crawled data reuse. A Semantic crawler based on extended CBR algorithm [9] refers to harvesting semantic web contents by crawler which abstract metadata from online web pages and cluster them by associating with ontological concepts which is based on CBR algorithm. Swoogle crawler [8] can harvest, parse and analyze semantic web documents or semantic web information pieces embedded in web documents. RDF crawler [2] which is a multithreaded implementation capable of downloading simultaneously from many sources while aggregation thread does the processing. It builds a model that remembers the provenance of RDF and takes care to delete and replace triples if it hits the same URL twice.

Based on the above literature it is found very limited work is done in semantic web crawlers that focuses on extracting information from ontology annotated documents that will help in better information retrieval. The proposed work is different from all these related work, as it is focused on crawling the semantic web documents annotated with ontology for harvesting the knowledge base to produce more inferential results. A novel framework is developed that works on ontology annotated web pages and extract the triples relation from the underlying ontologies associated with the web page.

4 Proposed Framework

SemCrawler is the proposed crawler having features that crawls the HTML pages annotated with RDF/OWL ontology. The proposed crawler incorporates a filter module that filters out the HTML web pages and crawls the documents annotated with ontologies. The proposed architecture is shown in Fig.1. The proposed framework consists of following functional components:

i. Fetch Module
ii. Filter Module
iii. Link Extraction Module
iv. URI Dispatcher Module
v. Parser Module

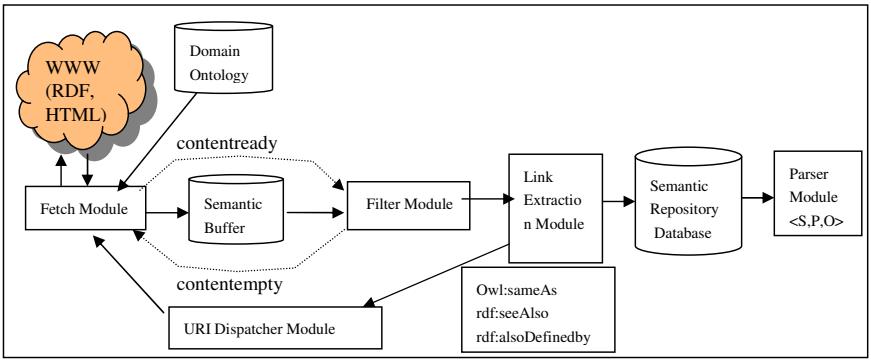

Fig. 1 Architecture of Proposed SemCrawl Framework

i. Fetch Module

It is a module that fetches HTML, RDF contents, corresponding ontologies associated with the web. After fetching, it stores, the web contents and ontologies in a semantic buffer. When the contents are transferred by fetch module to semantic buffer, signal "contentready" is sent to the filter module that filters the HTML web contents. On getting the "contentempty" signal from filter module, this module starts fetching the documents again from the World Wide Web.

```
Fetch Module
{
        wait (contentempty);
        extract URI from the URI queue;
        fetch the contents from web;
        store the contents in semantic buffer;
        signal (contentready);
}
```

ii. Filter Module

Filter module waits for the signal from the fetch module. After receiving the signal from fetch module, it gets the fetched contents stored in semantic buffer, filters the HTML web contents and the filtered contents are given as input to link extraction module for further processing sending the "contentempty" signal to fetch module for further fetching of contents from the web.

```
Filter Module (input: semantic buffer, output: filtered pages)
{
        wait (contentready);
        input the contents from buffer;
        extract the rdf web pages;
        input the filtered contents to link extraction module;
        signal (contentempty);
}
```

iii. Link Extraction Module

Link extraction module looks upon the contents which the filtered web contents got as output from the filter module. This module extracts certain constructs from the web page which works same as hyperlink <link href = " ">in HTML page representation, hence extracts the further links from the pages is then given as input to URI queue.

```
Link Extraction Module
{
if (Owl: sameAs or rdf: seeAlso or rdf: alsoDefinedby constructs in page)
input those links to URI queue for further processing;
else
store the web page in semantic repository database for further extraction of concepts;
}
```

iv. URI Dispatcher Module

This module gets the input from link extraction module, a list of URIs, which are given as input to fetch module for further downloading of semantic web documents from the web.

v. Parser Module

This module gets the input from semantic repositories database and will parse the web contents.

```
Parser Module
{
input the crawled data from semantic repository;
extract subject, predicate and object from the crawler output repositories;
store triples in database with three columns subject, predicate, object<S, P, O>;
}
```

5 Experiments and Results

For the purpose of first experiment 20 web pages were taken, out of which 5 web pages were HTML web pages and rest 15 pages were associated with ontology related to laptop domain and then gradually the number of pages were increased for subsequent tests as shown in Table 2. SemCrawl crawler was implemented in java using eclipse framework. Crawler was able to crawl all the web pages and ontology with filtering out the HTML web pages. This ontology annotation will help in finding relation between entities of the web page which will increase the further scope of research that the domain ontology developed by this research could be extended and used for classified and relevant results on the web.

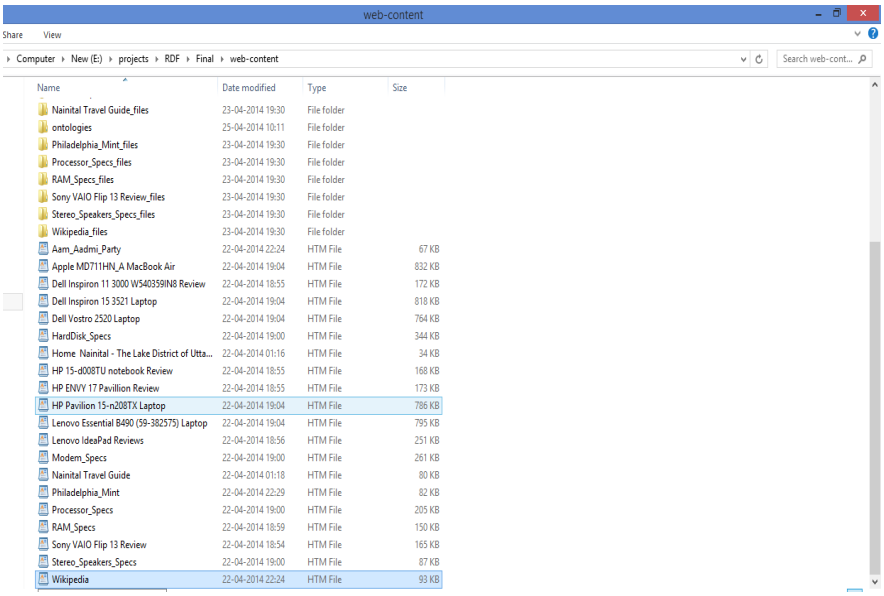

Fig. 2 Repositories of Web Pages

Fig.2. shows the repositories of web pages which contain plain HTML pages and HTML pages associated with ontology.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+RDFa 1.0//EN" "http://www.w3.org/MarkUp/DTD/xhtml-rdfa-1.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" version="XHTML+RDFa 1.0"

xmlns:laptop-reviews-ontology="http://www.semanticweb.org/vandana/ontologies/2014/1/laptop-reviews-ontology#">
<head>
<meta http-equiv="content-type" content="text/html; charset=UTF-8"><script src="Lenovo%20IdeaPad%20Reviews_files/beacon.js" async="" type="text/javascript">
    Lenovo IdeaPad Yoga 2 11 - Tablet Reviews
</title><link rel="shortcut icon" href="http://www.laptopmag.com/favicon.ico" type="image/x-icon">


    <!-- Pulled from http://code.google.com/p/html5shiv/ -->
    <!--[if lt IE 9]>
    <script src="//html5shim.googlecode.com/svn/trunk/html5.js"></script>
    <![endif]-->


    <link rel="stylesheet" type="text/css" href="Lenovo%20IdeaPad%20Reviews_files/main.css"><link rel="stylesheet" type="text/css" href="Lenovo%20IdeaPad%20


    <script gapi_processed="true" src="Lenovo%20IdeaPad%20Reviews_files/plusone.js" async="" type="text/javascript"></script><script src="Lenovo%20IdeaPad%2
    <script type="text/javascript" src="Lenovo%20IdeaPad%20Reviews_files/common_review.js"></script>
    <script src="Lenovo%20IdeaPad%20Reviews_files/search.js"></script>
    <script src="Lenovo%20IdeaPad%20Reviews_files/top_five.js"></script>
    <script src="Lenovo%20IdeaPad%20Reviews_files/fixed_top_bar.js"></script>
```

Fig. 3 Screenshot of a Web Page associated to Ontology

Fig.3. Shows an example of a web page associated with ontology. SemCrawl will crawl these pages associated with ontology and filter out HTML web pages with which no ontology is associated.

5.1 Crawler Evaluation

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
     xmlns:dcterms="http://purl.org/dc/terms/"
     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">

<rdf:Description rdf:about="file:/E:/projects/RDF/Final/web-
content/Dell%20Inspiron%20Seller.htm">
     <dcterms:title xml:lang="vi">Dell Inspiron
Sellor</dcterms:title>
</rdf:Description>

<rdf:Description
rdf:nodeID="noded3e3c44b27e282f69cdfd72b2b36f0bc">
     <rdf:type rdf:resource="http://schema.org/Product"/>
     <name xmlns="http://schema.org/Product/" xml:lang="vi">
                Dell Inspiron 15 3521 Laptop (3rd Gen Ci3/
4GB/ 500GB/ Linux)          </name>
</rdf:Description>

<rdf:Description rdf:nodeID="node9285a2090813a4561507dae4945778">
     <rdf:type rdf:resource="http://schema.org/Offer"/>
```

Fig. 4 Crawler Output of RDF Webpage

Crawler Output

The implemented system crawled all the RDF pages embedded with ontology from the repository and filter out HTML pages. Fig.4. shows the output of crawler, which will be all web pages that are associated with ontology. We have implemented our system with the series of three tests Test1, Test2 and Test3 with each test taking a collection of 20 pages, 50 Pages and 100 Pages respectively, With each test 100% accuracy was achieved for the module created, we were able to crawl all the relevant contents from the web discarding the unwanted HTML web pages. Table 2 indicates the results of the various test conducted on crawler module.

Table 2 Series of Test conducted on Repositories for Crawler Module

Test	Repositories	Repositories	Crawler Output
Test 1	20 Pages	RDF Web Pages	20 Pages
Test 2	50 Pages	RDF,OWL,HTML	38 Pages with filtered out 12 HTML pages
Test 3	100 Pages	RDF,OWL,HTML	85 Pages with filtered out 15 HTML pages

5.2 Parser Evaluation

Parser modules have been implemented in java using eclipse framework and jena API library. Jena is convenient toolkit to manipulate RDF models for developing application within semantic web [3]. Parser module extracts triples from the semantic database repositories. Triples are in the form <Subject, Predicate, Object>. Number of triples depends on the vocabulary of our ontology. Output of triples extracted of parser module developed by us has been specified in Table 2. and Fig. 5 shows the output of parser module showing the triples relation<S, P, O>.Table 3 indicates the result of triples discovered by parser module.

```
**********************************
Subject       : http://www.semanticweb.org/vandana/ontologies/2014/1/laptop-reviews-ontology
Predicate     : http://www.w3.org/1999/02/22-rdf-syntax-ns#type
Object        : http://www.w3.org/2002/07/owl#Ontology
**********************************
Subject       : http://www.semanticweb.org/vandana/ontologies/2014/1/laptop-reviews-ontology#Laptop_Advisors
Predicate     : http://www.w3.org/2000/01/rdf-schema#subClassOf
Object        : http://www.semanticweb.org/vandana/ontologies/2014/1/laptop-reviews-ontology#Laptop
**********************************
Subject       : http://www.semanticweb.org/vandana/ontologies/2014/1/laptop-reviews-ontology#Laptop_Advisors
Predicate     : http://www.w3.org/1999/02/22-rdf-syntax-ns#type
Object        : http://www.w3.org/2002/07/owl#Class
**********************************
Subject       : null
Predicate     : http://www.w3.org/1999/02/22-rdf-syntax-ns#rest
Object        : http://www.w3.org/1999/02/22-rdf-syntax-ns#nil
**********************************
Subject       : null
Predicate     : http://www.w3.org/1999/02/22-rdf-syntax-ns#first
Object        : http://www.semanticweb.org/vandana/ontologies/2014/1/laptop-reviews-ontology#NDTVTechnoWorld
**********************************
```

Fig. 5 Console output of parser module extracting Subject, Predicate, Object<S, P, O> Triples

Table 3 Triples discovered by Parser Module

Number of Web pages crawled	20
Number of Triples discovered	146 Triples

Crawler and parser module were executed on particular repository and RDF triples.

6 Conclusion

This paper presented a detail insight into the various differences between crawling in the traditional web as compared to crawling in the semantic web documents. Also the technique with which semantic data can be crawled, for later indexation and classification is proposed. In this research, crawler framework is implemented for harvesting web pages associated with ontology and creating semantic web based knowledge base. In future work further study on the different indexing mechanism of the ontologies associated with the web pages will be done. This paper is concluded with that there is great potential for research in this area regarding how these documents can be indexed which is a very innovative concept.

References

[1] Berners-Lee, T., Hendler, J., Ora, L.: The Semantic Web. Scientific American 284(5), 34–43 (2001)
[2] Biddulph, M.: Crawling the Semantic Web. BBC London, United Kingdom (2003)
[3] McBride, B.: Jena implementing the RDF Model and Syntax Specification. Hewlett Packard laboratories, Bristol, UK (2000)
[4] DARPA Agent Markup Language (2012), http://www.daml.org/language/
[5] Dhingra, V., Bhatia, K.K.: Towards Intelligent Information retrieval on Web. IJCSE (2011)
[6] Dhingra, V., Bhatia, K.K.: Metadata: Towards Machine-Enabled Intelligence. IJWesT 3(3), 121–130 (2012)
[7] Dodds, L.: Slug: A Semantic Web Crawler in Jena User Conference Bristol, UK (2006)
[8] Li, D., et al.: Swoogle: A search and metadata engine for the semantic web. In: Proceedings of 13th ACM Conference on Information and Knowledge Management (2004)
[9] Dong, H., Hussain, F.K., Chang, E.: A semantic crawler based on an extended CBR algorithm. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2008 Workshops. LNCS, vol. 5333, pp. 1076–1085. Springer, Heidelberg (2008)
[10] Asunción, G.-P., Corcho, O.: Ontology Languages for the Semantic Web. IEEE Intelligent Systems Journal (2002)
[11] Andreas, H., Hannes, G.: On Searching and Displaying RDF Data from the Web. In: 2nd European Semantic Web Conference (ESWC 2005), Heraklion, Greece (2005)
[12] Hendler, J., Berners-Lee, T.: From Semantic Web to Social Machine. Artificial Intelligence 174(2), 156–161 (2010)
[13] Vishal, J.: Ontology Based Information Retrieval in Semantic Web. International Journal of Information technology and Computer Sceince, 62–69 (2013)

[14] Annett, M., Ronny, W., Klaus: Searching Community-built Semantic Web Resources to Support Personal Annotation. In: Proceedings of Bridging the Gap between Semantic Web and Web 2.0, Austria (2007)
[15] Van de Maele, F., Spyns, P., Meersman, R.: An Ontology-Based Crawler for the Semantic Web. In: Meersman, R., Tari, Z., Herrero, P. (eds.) OTM 2008 Workshops. LNCS, vol. 5333, pp. 1056–1065. Springer, Heidelberg (2008)
[16] Staab, S., Apsitis, K., Handschuh, S., Oppermann, H.: Specification of an RDF Crawler (2004)
[17] World Wide Consortium RDF Primer, http://www.w3.org/TR/rdf-primer/