A SEMANTIC BASED APPROACH FOR INFORMATION RETRIEVAL FROM HTML DOCUMENTS USING WRAPPER INDUCTION TECHNIQUE

A.M.Abirami[1], Dr.A.Askarunisa[2], T.M.Aishwarya[1] and K.S.Eswari[1]

[1]Department of Information Technology,
Thiagarajar College of Engineering, Madurai, Tamil Nadu, India.
abiramiam@tce.edu, {aish2292,eswainnov}@gmail.com
[2]Department of Computer Science and Engineering,
Vickram College of Engineering, Enathi, Tamil Nadu, India.
nishanazer@yahoo.com

ABSTRACT

Most of the internet applications are built using web technologies like HTML. Web pages are designed in such a way that it displays the data records from the underlying databases or just displays the text in an unstructured format but using some fixed template. Summarizing these data which are dispersed in different web pages is hectic and tedious and consumes most of the time and manual effort. A supervised learning technique called Wrapper Induction technique can be used across the web pages to learn data extraction rules. By applying these learnt rules to web pages, enables the information extraction an easier process. This paper focuses on developing a tool for information extraction from the unstructured data. The use of semantic web technologies much simplifies the process. This tool enables us to query the data being scattered over multiple web pages, in distinguished ways. This can be accomplished by the following steps – extracting the data from multiple web pages, storing them in the form of RDF triples, integrating multiple RDF files using ontology, generating SPARQL query based on user query and generating report in the form of tables or charts from the results of SPARQL query. The relationship between various related web pages are identified using ontology and used to query in better ways thus enhancing the searching efficacy.

KEYWORDS

Information Retrieval, Ontology, Structured Information Extraction, RDF, SPARQL, Semantic Web.

1. INTRODUCTION

Today's web contents are suitable only for humans; it is not machine understandable and readable. So retrieving information from web pages is a complex task. Time consumption is more if one needs to refer or access too many HTML pages to collect data and make summary on them.

Jan Zizka (Eds) : CCSIT, SIPP, AISC, PDCTA - 2013
pp. 509–516, 2013. © CS & IT-CSCP 2013 DOI : 10.5121/csit.2013.3655

The semantic web technologies [16] overcome these difficulties to a greater extent. A core data representation format for semantic web is Resource Description Framework (RDF). RDF is a data model for web pages. RDF is a framework for representing information about resources in a graph form. It was primarily intended for representing metadata about WWW resources, such as the title, author, and modification date of a Web page, but it can be used for storing any other data. It is based on triples subject-predicate-object that form graph of data [18].

All data in the semantic web use RDF as the primary representation language [9]. RDF Schema (RDFS) can be used to describe taxonomies of classes and properties and use them to create lightweight ontologies. Ontologies describe the conceptualization, the structure of the domain, which includes the domain model with possible restrictions [11]. More detailed ontologies can be created with Web Ontology Language (OWL). It is syntactically embedded into RDF, so like RDFS, it provides additional standardized vocabulary. For querying RDF data as well as RDFS and OWL ontologies with knowledge bases, a Simple Protocol and RDF Query Language (SPARQL) is available. SPARQL is SQL-like language, but uses RDF triples and resources for both matching part of the query and for returning results [10].

Ujjal Marjit et.al. [1] presented a linked data approach to discover resume information by providing a standard data model, data sharing and data reusing. Arup Sarkar et.al. [2] provided an approach to publish the data from legacy databases as linked data on the web thus enabling the machine readability. David Camacho et.al. [3] extracted information from HTML documents and added semantics through XML to generate rules using Web Mantic with the help of user interaction. Uriv shah et.al [4] provided a framework for retrieval of documents containing both free text and semantically enriched markup using DAML+OIL semantic web language. Olaf Hartig et.al. [5] developed an approach to execute SPARQL queries over web of linked data to discover relevant data for query answering during query execution itself using RDF links between data sources. Peter Haase et. al. [6] assumed a globally shared domain ontology, which can not only be used for subsequently classifying the bibliographic metadata, but also for supporting an improved query refinement process. Abirami et. al [7] proposed a Semantic based approach for generating reports from HTML pages by converting those HTML pages into pre-processed and formatted CVS files, generating OWL files for domain, separating contents from CVS files based on OWL files and rules to generate RDF files and querying RDF files using SPARQL.

This paper proposes a model for storing the content of HTML pages into RDF format using the rules generated by the wrapper induction technique. For example, the university web site may contain the details of all the staff members of all the departments mostly in HTML pages. The data is collected and consolidated mostly manually by every year. Though the content is readily available in HTML pages, the manual effort is wasted while consolidating the report. The proposed model generates the report from different HTML documents using RDF and SPARQL. Only the relevant information is extracted and stored in RDF format which makes further querying easier. This paper is organized into following sections: section 2 describes the methodology, section 3 shows the experiments that are taken and section 4 gives the conclusion.

2. METHODOLOGY

The proposed model includes five different processes namely – (i) extracting structured data from multiple HTML pages (ii) generating RDF files with the structured data extracted (iii) creating ontology for specific domain relationship (iv) generating corresponding SPARQL queries for the

user query and (v) generating reports according to the user needs. The Figure 1 shows the various processes in the implementation, which are dealt in the subsequent sections.

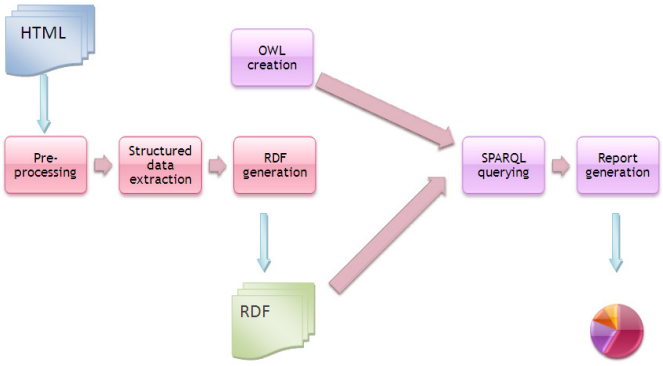

Figure. 1. Model for Information Retrieval

2.1 Pre-Processing of HTML pages

It is important to check for the clean HTML pages, because a typical Web page contains a large amount of noise, which can adversely affect data extraction accuracy. Pre-Processing phase is done before actual extraction starts, which is depicted in Figure 2.The raw HTML is fed into the HTML Tidy [13]. This tool checks for the presence of close tag for every open tag. The resultant well formed html is fed into noise removal phase, where the output will be a clean HTML page.

2.2 Structured Data Extraction & RDF Generation

Once the pre-processing phase gets completed, the clean html pages are fed into structured data extraction phase. This is accomplished by using the Wrapper Induction Technique [17], which is the supervised learning approach, and is semi-automatic. In this approach, a set of extraction rules is learned from a collection of manually labeled pages. The rules are then employed to extract target data items from other similarly formatted pages. The process is pictorially represented in Figure 3.

Most HTML tags work in pairs. Each pair consists of an open tag and a close tag indicated by < > and </> respectively. Within each corresponding tag-pair, there can be other pairs of tags, resulting in nested structures. Thus, HTML tags can naturally encode nested data. Notable points are: (a) there are no designated tags for each type as HTML was not designed as a data encoding language and any HTML tag can be used for any type (b) for a tuple type, values (data items) of different attributes are usually encoded differently to distinguish them and to highlight important items. In case of any conflicts, they are identified and corrected by replacing or adding rules. These two processes go iteratively. Once the resultant structured data goes along with the human oracle, the extracted structured data is sent for the RDF generation phase.

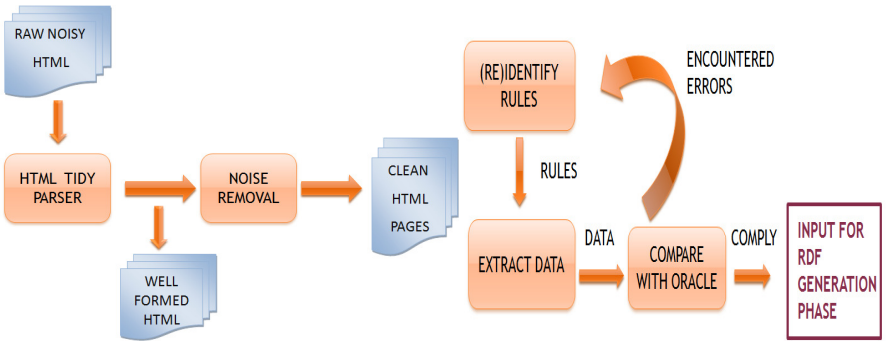

Figure. 2. Pre-processing of web pages

Figure. 3. Structured data extraction

The extraction is done using a tree structure called the *EC* tree (embedded catalog tree) [17], which model the data embedding in a HTML page. For example EC tree used in this paper is given in Figure 4.The wrapper [18] uses a tree structure based on this to facilitate extraction rule learning and data extraction. Given the *EC* tree and the rules, any node can be extracted by following the tree path *P* from the root to the node by extracting each node in *P* from its parent. The extraction rules are based on the idea of landmarks. Each landmark is a sequence of consecutive tokens and is used to locate the beginning or the end of a target item.

Some of the rules identified by the tool are:

- HTML page with the word “journal” shows that the faculty has published journals.
- Pattern says that the very first list (“ ”) is the Journal list.
- in a list (<li>) within journal list, the contents before the bold tag(<b>) are Authors names.
- content within the bold tag is the Title of the paper published.
- content after the bold tag is the Journal name.
- within the journal name, the four consecutive numbers within the range, 19** to 20** is year of journal publication.
- Some of the author names are within bold tag, which gets included wrongly in journal name during extraction. After careful assessment, an “<auth> </auth>” tag is inserted between individual author names. During extraction, the presence of “auth” tag within bold tag is examined to extract the author names and title precisely.

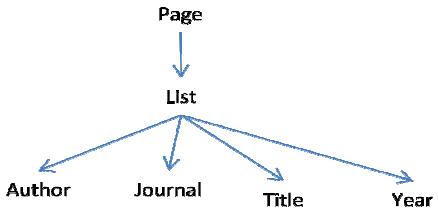

Figure. 4.Embedded catalog tree

Those rules were used to extract the structured data from the HTML pages of same format. The output is checked against the human oracle. In case of conflict, the conflicting things were identified and corrected by replacing or adding rules. These two processes go iteratively. Once the resultant structured data goes along with the human oracle, the extracted structured data is sent for the RDF generation phase. With the default format of the RDF file and the structured data extracted, the RDF file is generated for its corresponding HTML file.

2.3 OWL Creation

Domain specific vocabularies and the relationships are established by ontology creation step using the Protégé tool [8]. The relationship between those RDF files is established by .owl file [12]. The ontology developed by this step is very much useful during the report generation by correctly identifying the RDF file for search. SPARQL query [9] is analyzed and the report is generated in the required format using JENA APIs [11]. This is further explained in the subsequent section.

3. EXPERIMENTAL RESULTS

The proposed model is experimented with the college web site (www.tce.edu) to get the report on the journal publications of various departments' staff members for a particular time period. HTML pages containing Journal details are pre-processed. Some faculty members may not have updated their details in HTML page. Such HTML pages are rejected before pre-processing phase.

Table 1. Experimental data considered

Dept	# Html Pages	# Preprocessed Pages
CSE	28	11
ECE	32	26
EEE	30	14
MECH	34	18
CIVIL	23	14
IT	20	11

3.1 HTML to RDF conversion

Each cleaned HTML pages are converted into RDF [9] as shown below in Figure 5.

```
<rdf:Description>
 <staff:authors> A.M.Abirami  </staff:authors>
 <staff:title> An Enhanced ….. </staff:title>
<staff:journal>International Journal …. </staff:journal>
<staff:year>2012</staff:year>
</rdf:Description>
```

Figure. 5. RDF for HTML page

3.2 Evaluation Metrics

We've considered precision as a measure [17] before and after the rule refinement. Precision is calculated as the ratio of information retrieved and the total information. The precision value is measured against the number of correctly extracted information under the right RDF tag for 94 HTML pages. The following Table 2 gives the precision values for various tags.

Table 2. Precision for RDF conversion

Extraction of	Before Rule Refinement	After Rule Refinement
Author	51% (48)	89% (84)
Title	75% (71)	94% (89)
Journal	100% (94)	100% (94)
Year	68% (64)	88% (83)

Lower precision in extracting author names is due to improper alignment of author names which cannot be corrected in preprocessing phase. Precision is increased by rule refinement for the following problems that are dealt: many consecutive commas, misplaced commas and absence of comma. The following corrective actions are also taken: (i) Within the journal name, the four consecutive numbers within the range, 19** to 20** is considered as year (ii) Some of the author names are within bold tag, which is wrongly placed, thus adding <auth> in between the author names.

3.3 Query Results

For an instance, a general query [15] for retrieving all details is given below.

```
rdf:http://www.w3.org/1999/02/22-rdf-syntax-ns#
PREFIX staff:http://www.w3schools.com/rdf/
SELECT ?authors ?title ?journal ?year
WHERE {?person staff:authors ?authors.
?person staff:?title.
?person staff:journal ?journal.
?person staff:year ?year }
```

3.4 Report Generation

One of the ways of presenting the output is through various charts. The tool uses JFreeChart [14] and enables easy query interface for various factors and they are depicted in the series of Figures 6-9.

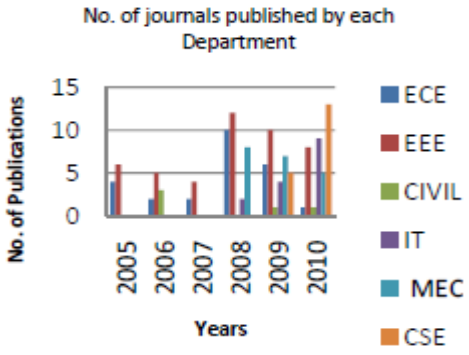

Figure.6. Number of journals published by each department in the range of years

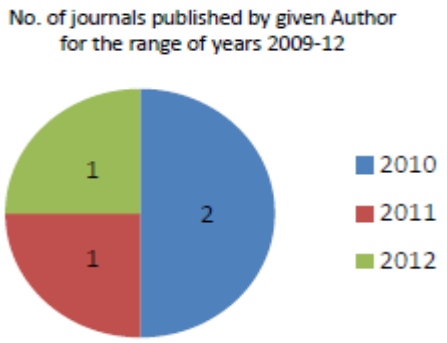

Figure.7. Number of journals published by given author in given range of year

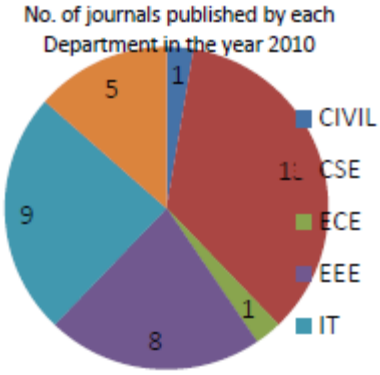

Figure. 8. Number of journals published by each department in given year

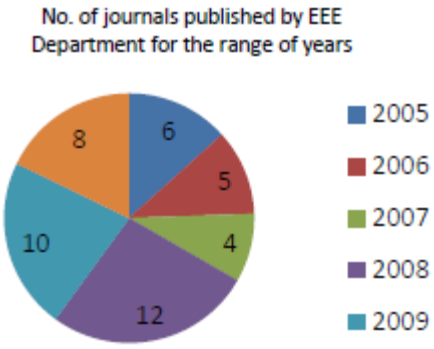

Figure. 9. Number of journals published by given department in the range of years.

Time taken by manual effort includes time taken to traverse the links to access required HTML document, scan the page for required details, identify and feed the details in different location (eg. spreadsheet) and generate report. These functions takes time in the order of minutes and in worst case it takes hours, based on the person doing the job. But the toolkit enables to update the repository on regular basis. Once the details are updated, required information can be obtained and reports can be generated in seconds. Time taken by tool includes - generation of SPARQL query, identifying required RDF files from ontology, extracting details from RDF files identified and generating the report. The time taken to generate the reports is shown in Table 3.

Table 3. Report generation time

Report No.	Manual Effort (in min)	Toolkit's time (in sec)
R1	1416	4
R2	184	3
R3	670	4
R4	14	2

4. CONCLUSION

We developed a toolkit to retrieve the web of data using RDF and ontology by well refined rules to extract the data. Quick and useful inferences can be made by easy query building and report generation process. As a future enhancement, inter-department relationships can be made using MULTIPLE ontologies to bring out the inter-college relationships.

REFERENCES

[1] Ujjal Marjit,Kumar Sharma and Utpal Biswas,"Discovering Resume Information using Linked data", Information Journal of web & semantic technology(IJWesT) Vol.3,No.2,April 2012.

[2] Arup Sarkar,Ujjal Marjit and Utpal Biswas,"Linked data generation for the University data from Legacy database", International Journal of Web & Semantic technology(IJWesT) Vol.2,No.3,July 2011.

[3] David Camacho and Maria D.R-Moreno,"Web data Extraction using Semantic Generators", VSP International Science Publishers,2006.
[4] Uriv shah,Tim Finin,Anupam Joshi,R.Scott cost,James Mayfield,"Information Retrieval on the Semantic web".
[5] Olaf Hartig,Christian Bizer and Johann-Christoph Freytag ,"Executing SPARQL Queries over the Web of Linked Data".
[6] Peter Haase, Nenad Stojanovic, York Sure, and Johanna Volker, “Personalized Information Retrieval in Bibster,a Semantics-Based Bibliographic Peer-to-Peer System”
[7] A.M.Abirami , Dr.A.Askarunisa, ”A Proposal for the Semantic based Report Generation of Related HTML Documents”, International Journal of Software Engineering and Technology, Sept 2011.
[8] http://protege.stanford.edu/
[9] http://www.w3schools.com/rdf/rdf_example.asp
[10] http://www.w3.org/TR/rdf-sparql-query/
[11] http://jena.sourceforge.net/tutorial/RDF_API/
[12] http://www.obitko.com/tutorials/ontologies-semanticweb/ontologies.htm
[13] http://tidy.sourceforge.net/
[14] http://www.jfree.org/jfreechart/
[15] http://code.google.com/
[16] http://www.ai.sri.com/~muslea/PS/jaamas-2k.pdf
[17] Bing Liu,"Web data mining - Exploring hyperlinks,contents,and usage data".
[18] Grigoris Antoniou and Frank van Harmelen, "A semantic Web Primer".