A Novel Technique for Creating Semantic Database for Hidden Web

Manvi
Department of Computer Engineering
YMCA University of Science & Technology Faridabad, India

Ashutosh Dixit
Department of Computer Engineering
YMCA University of Science & Technology Faridabad, India

Komal Kumar Bhatia
Department of Computer Engineering
YMCA University of Science & Technology Faridabad, India

Rinki Kardam
Department of Computer Engineering
YMCA University of Science & Technology Faridabad, India

Abstract

Today a lot of valuable information on WWW lie behind the search forms. This information is known as Hidden Web. To extract this information user has to fill various forms with appropriate values. Traditionally user has to type in these values manually. For automatically filling these types of interfaces precisely relevant information is needed. A database that stores semantic information about objects and their relations may solve this problem. This database can be defined with the help of Ontology which defines common vocabulary. With the help of RDF attached with web pages, ontology can be created and stored in database. The accurate and meaningful information retrieved from RDF of web pages can be stored and used later for filling up above form interfaces. In this work, a database that will help user to fill the form automatically with the values has been created with the help of ontology for book domain.

Keywords: Hidden Web, Domain Specific, Ontology, Database RDF

I. INTRODUCTION

WWW contains a large amount of information that can be categorized into two different sets ie: Surface Web and Hidden Web. The Surface Web consists of interlinked pages, while Hidden Web contains information in the form of data sources. The surface web contains hyperlinks hence can be crawled and indexed by general purpose search engines, while the hidden Web refers to the information that is "hidden" behind the query interfaces and is not directly accessible to search engines. A Hidden Web crawler produces results which are behind the search interfaces or forms, dynamically retrieved in response to user query. There is no keyword matching scheme and no url following for accessing hidden web data as depicted in figure 1.

A lot of research is going on to devise new methods for indexing and accessing hidden web. Retrieving hidden web mainly consists of following major tasks [5]:

1) Making common interface.
2) Creating database/repository for finding values for interfaces.
3) Filling values into the interface fields
4) Generating queries.

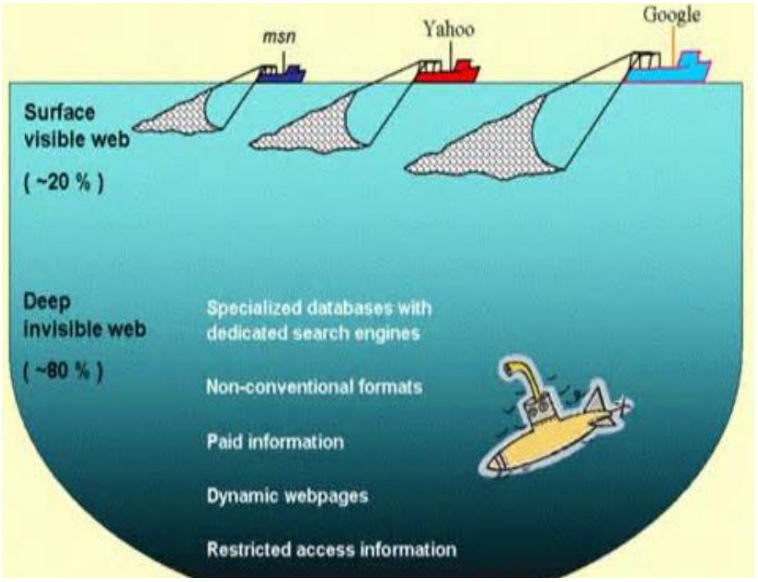

Fig. 1: Hidden Web

This paper concentrates on the second part i.e. creation of database. Till now very few techniques to deal with this database, have been implemented but the use of ontology for the creation of database to retrieve hidden web content has not been implemented yet.

Need of Ontology is aroused which Classify each searchable form to its relevant domain then match the suitable ontology to automatically fill out these forms. With the help of RDF attached with web pages, ontology can be created and stored in database. This accurate and meaningful information can be retrieved and used for filling up above said web interfaces.

Ontology also defines the concepts about Web page categories and their hierarchical relationships. In almost every ontology, concepts are described by terms. Note that each concept might have more than one term describing it and that a term need not match only one concept. For example, to describe the concept of bicycle the terms "bicycle" and "bike" can be used. However, the term "bike" might also refer to the concept of motorcycle. Usually, ontologies include a single and unambiguous term for each concept.

Figure 2 shows an example of book ontology manually created by us in Protégé [6].

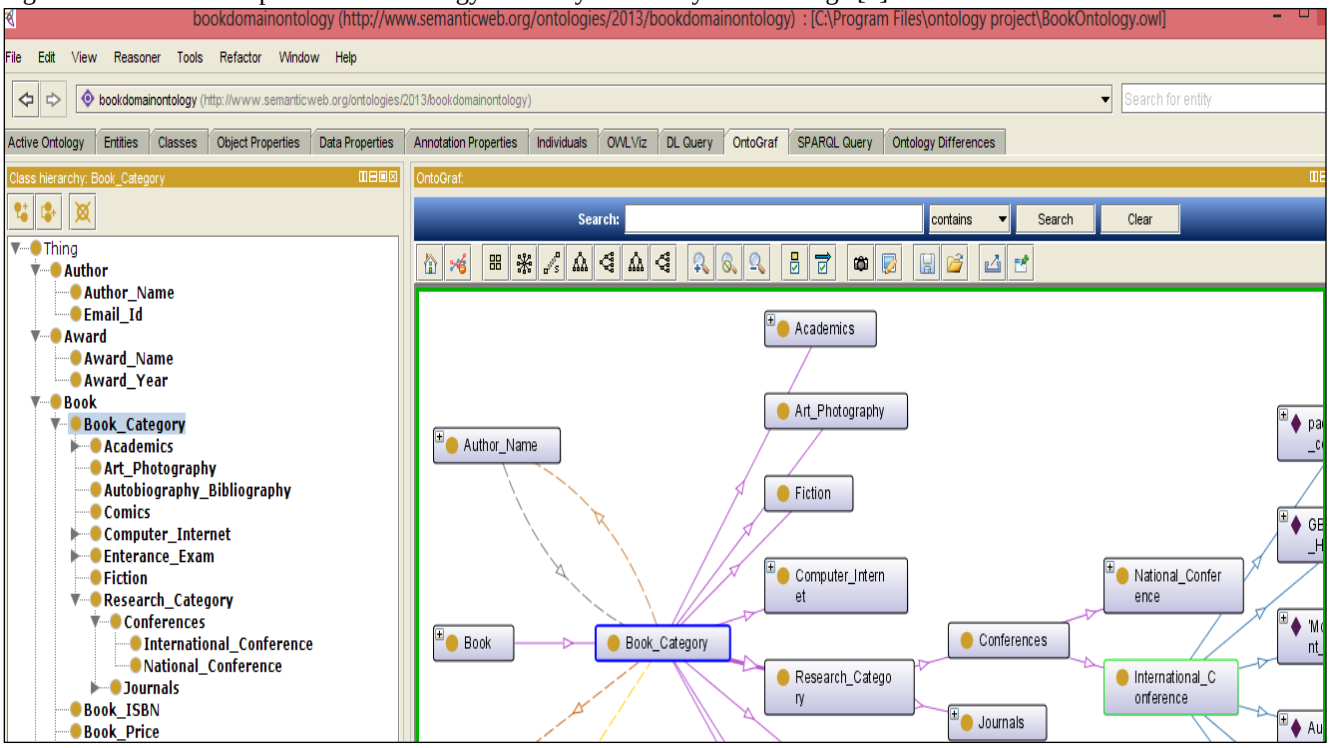

Fig. 2: Book Ontology

II. LITERATURE SURVEY

The size of this hidden web has been estimated around 500 times the size Surface web [1]. As the volume of hidden information is on a fast pace, researchers have increased their interest to work in this field, hence constant work to find out techniques that allow users and applications to leverage this information is going on.

Few of the wok done among them is discussed below:

A. Hidden Web Exposer (HiWE):

by Sriram Raghavan and Hector Garcia-Molina [2]: In this approach, a model of task specific human assisted web crawler called HiWE was designed to automatically process, analyze, and submit forms, using an internal model of forms and form submissions. It uses a layout-based information extraction (LITE) technique to process and extract useful information. this architecture allows the crawler to concentrate on relevant pages only but not precise with regards to partially filled forms.

B. Collecting Hidden Web Pages for Data Extraction:

Juliano Palmieri Lage et al. [10]: In this paper, a concept of web wrappers which extracts the unstructured data from web pages. It takes a set of target pages generated by an approach called "Spiders" which automatically traverse the web for web pages. The advantage of this technique is that it can access a large number of web sites of diverse domains. The limitation of this technique is that it can access only that web site that follows common navigation patterns.

C. Understanding Web Query Interfaces [5]:

Zhen Zhang et al. addresses the problem of understanding web search interfaces by presenting a best-effort parsing framework. This work presented a form extractor framework based on 2P grammar and the best effort parses in a language parsing

framework. It identifies the search interface by continuously producing fresh instances by applying productions until attaining a fix-point, when no fresh instance can be produced. Best effort parser technique minimizes wrong interpretation as much as possible in a very fast manner. Limitation of this technique was that establishment of single global grammar was required that can be interacted to the machine globally.

D. Automated Discovery of Search Interfaces on the web [7]:

In this papar Jared Cope, Nick Craswell and David Hawking defined a novel technique to automatically detect search interface from a group of html forms. A decision tree was developed with the C4.5 learning algorithm using automatically generated features from html mark up that can give a classification accuracy of about 85% for general web interfaces. This technique is helpful to automatically discover the search interface but it is based on single tree classification method and number of feature generation is limited due to use of limited data set.

III. CHALLENGES

After having a critical look over the work done in this field and considering the limitation of each, It is observed that Hidden Web services which are mostly form-based interfaces need to be described using new approach that may involve attaching meaning to the data. All the parameters involving functional as well as non-functional properties need to be extracted or inferred.

1) It is noticed that various researchers have given algorithms to access deep web using various techniques for automating the task of extraction of form pages and crawling form pages but none has described the way to automatically find the values that need to be filled in the form page.
2) Also no method till now has been devised to create and populate the values in a database so that more and more values are available to fill various forms.
3) Basic keyword or label matching techniques results in to various error pages and irrelevant pages downloaded by hidden web crawler.

To get the relevant and more quality information the values which need to be filled in the form elements must be accurate and semantically correct. Ontology may solve this purpose well as it is a structured way of describing knowledge.

IV. PROPOSED WORK

From the above stated issues and literature survey it can be concluded that Hidden Web can't be indexed by conventional search engines. For generating and responding to user query in hidden web environment we need a database that is populated knowledge base which satisfies the user's need of information.

As user specifies queries using different keywords but having same meaning, Ontology is one of the best ways for creating Domain knowledge that has common understanding of the structure of information among people.

With the use of domain knowledge/semantics more hidden web pages can be extracted which contain high quality data. Also there should be an automatic way of creating ontology and store that ontology in a database (in form of tables) whose values can be retrieved when required. Ontology can be attached with a web page in the form of RDF tuples <Subject,Predicate,Object>.

Here a database using ontology has been constructed here which is contextual or semantically rich, which can be used to fill various label values in a search form.

With the help of RDF attached with web pages, ontology can be created and stored in database. This accurate and meaningful information can be retrieved and used for filling up above said web interfaces.

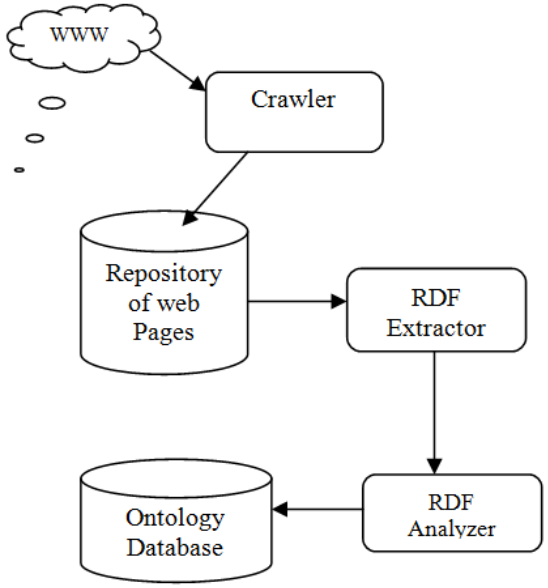

Fig. 3: Architecture of proposed System

The proposed architecture contains these five basic modules:

1) Crawler
2) Repository
3) RDF Extractor
4) RDF Analyzer
5) Database of Ontology

A. Crawler:

Crawler is a basic crawler which starts with a seed URL and downloads the web pages with the help of various links available in the web page in basic BFS manner. The crawler in this architecture is domain specific that only downloads the pages of book domain. These crawlers after downloading the pages store them in repository.

B. Repository of Web Pages:

This repository will contain all the web pages downloaded by crawler.

C. RDF Extractor:

This module will find out whether the web page contains RDF or not. If the downloaded page contains RDF then the page is sent to next module, but if page does not contain any RDF then that particular page is discarded.

D. RDF Analyzer:

This module is domain specific; which will analyze the rdf of pages for book domain and will save the particular RDF of pages in protégé as ontology as well as will send to next module to save in relational database.

E. Ontology Database:

This is a relational database created for book domain having various fields corresponding to the attributes found in RDF of web Pages. Initially this database was made manually then integrated with the implemented framework.

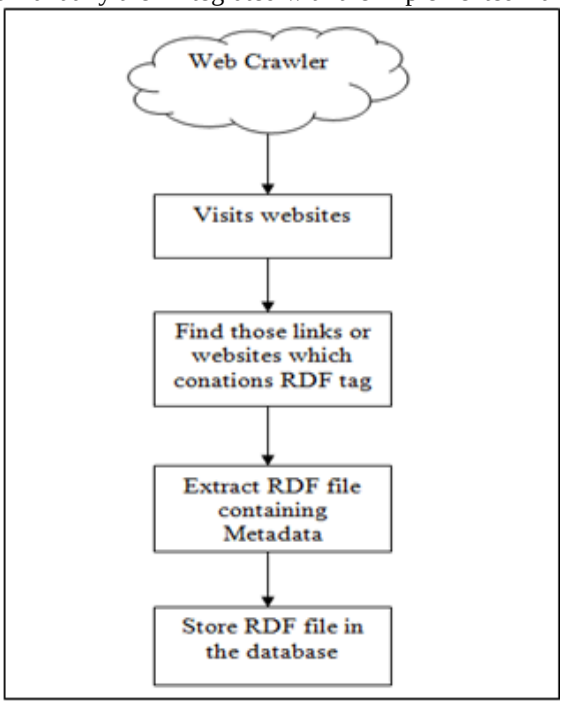

Fig. 4:.Basic Flow diagram of the system

Figure 4 represents the basic flow of the entire system.It can be seen that crawler after visiting a particular site downloads the pages.Then from all the pages only those pages are considered and sent further which contains RDF.The pages which donot conatain RDF tag are discarded.

V. IMPLEMENTATION AND RESULTS

In figure 5, a search form for extracting RDF is developed. This form consists of a textbox, a label and a search button. When a keyword is entered in the search text box and click on the search button. Then all the links and URLs whether it contain RDF or not will be retrieved.

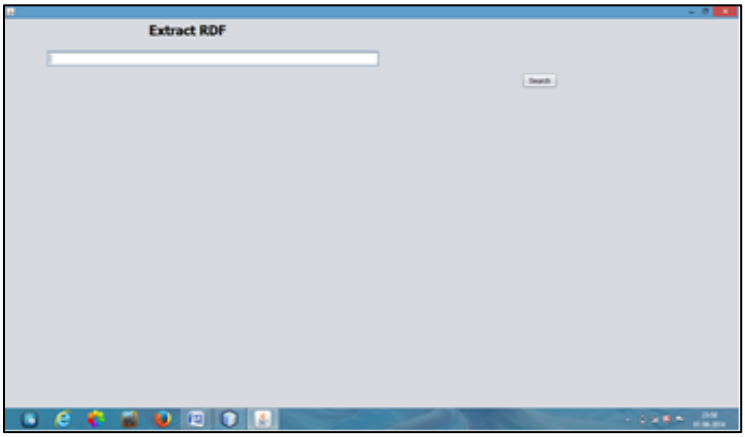

Fig. 5: Extraction of RDF

In Figure 6 when the user enters a query in the search text box then various URLs containing the book as keyword are retrieved. There is a list of URLs in which some URLs contain RDF or OWL file which can be saved and viewed in Protégé.

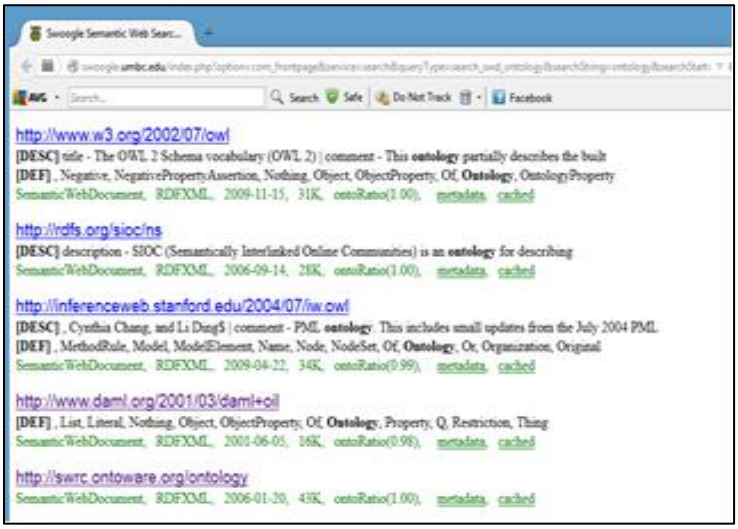

Fig. 6: Retrieving Various Links

Figure 7 shows the RDF structure of the particular page on which user clicks and save in the form of OWL document. This RDF file contains various Tags and Meta information which define various concepts and relationships (Ontology).

Fig. 7: RDF file corresponding retrieved link

In Figure 8. an interface is designed which contains a search box, a button and a text area with various attribute values. , a path of OWL file is specified in search text box and click on GO button. By clicking on button values will be stored this data

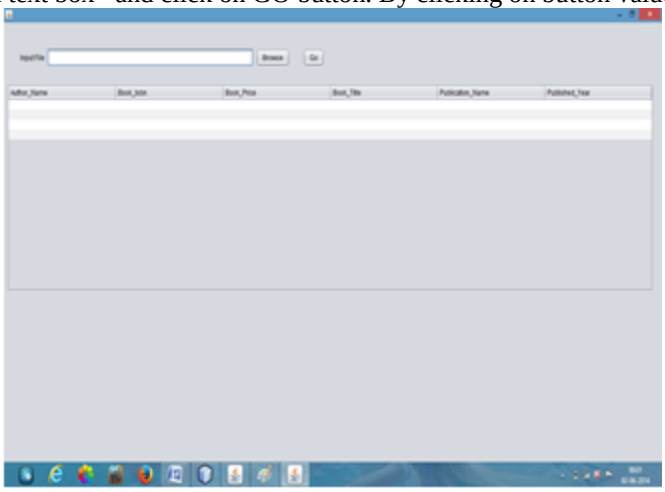

Fig. 8: Database interface

Figure 9 shows the actual values stored in database from the web page downloaded above. These values will be used further by a hidden web crawler to be filled in form interface.

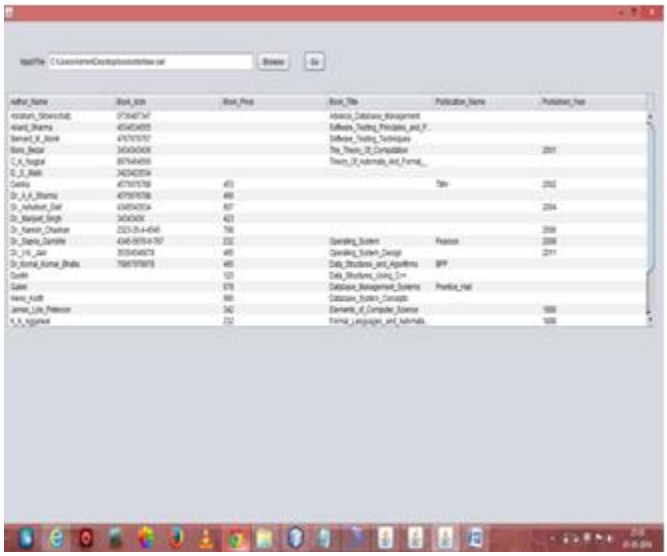

Fig. 9: Database containing value of various attributes

VI. CONCLUSION AND FUTURE SCOPE

In this paper, a technique of creating a searchable database for the hidden web content using Ontology from the web pages which contain RDF is proposed. This database will be helpful to deal with the search interface means the values that are stored in database will be used to fill the form that lie on the hidden web. After filling the various labels on a search form with appropriate values that are stored in the database higher quality information will be retrieved. Although this work makes the data searchable from the hidden web by creating a database for the Hidden web, this work does not solve all the purposes as WWW does not contain the RDF pages only .Also the method devised above uses relational database, to make full utilization of semantic information one must use semantic database available in Oracle 10 or 11 g.

REFERENCES

[1] DeepWeb.http://en.wikipedia.org/wiki/Deep web.
[2] Bin He, Mitesh Patel, Zhen Zhang, and Kevin ChenChuan Chang. Accessing the deep web. Communications of the ACM, 50(5):94{101, 2007.
[3] Komal Kumar Bhatia, A.K.Sharma, "A Framework for an Extensible
[4] Domain-specific Hidden Web Crawler (DSHWC)", communicated to IEEE TKDE Journal Dec 2008.
[5] Sriram Raghavan and Hector Garcia-Molina. Crawling the hidden web. In Proceedings of the International Conference on Very Large Data Bases, pages 129{138, San Francisco, CA, USA, 2001.
[6] Bergman, M.K. (2001). The Deep Web: Surfacing Hidden Value. In The Journal of Electronic Publishing, Vol. 7, No. 1.
[7] Manvi et al.,"Design of Ontology based adaptive Hidden Web Crawler", CSNT 2013,IEEE International Conference.
[8] Manvi et. al."Generating Domain Specific Ontology for hidden wen", ISCON 2014,IEEE International Conference.

[9] Z. Zhang, B. He, and K. Chang. Understanding Web Query Interfaces: Best-Effort Parsing with Hidden Syntax. In SIGMOD Conference , 2004.
[10] Raghavan, S. and Garcia-Molina, H. (2001). Crawling the Hidden Web. In Proceedings of the 27th International Conference on Very Large Data Bases, Roma, Italy.
[11] Cope, J., Craswell, N., and Hawking, D. (2003). Automated Discovery of Search Interfaces on the web. In Proceedings of the Fourteenth Australasian Database Conference (ADC2003), Adelaide, Australia.
[12] Barbosa, L., and Freirel, J.(2004). Siphoning Hidden-Web Data through Keyword-Based Interface., In Proceedings of SBBD.
[13] Lage, P. et al. " Collecting Hidden Web Pages for Data Extraction". In Proceedings of the 4th international workshop on Web information and data management 2002, PP: 69-75.
[14] Deng, X. B., Ye, Y. M., Li, H. B., & Huang, J. Z. (2008). An Improved Random Forest Approach For Detection Of Hidden Web Search Interfaces. In Proceedings of the Seventh International Conference on Machine Learning and Cybernetics, Kunming, China.
[15] Ye, Y., et al. (2009). Feature Weighting Random Forest for Detection of Hidden Web Search Interfaces. In Computational Linguistics and Chinese Language Processing , Vol. 13, No. 4, PP: 387-404.