# Hidden Web Query Technique for Extracting the Data From Deep Web Data Base

Nripendra Narayan Das[1], Ela Kumar[2]

**Abstract - Web databases are now present everywhere . The data from the Deep Web can not be accessed by Search engine and web crawlers directly. The only way to access the hidden database is through query interfaces and filling up number of HTML forms for a specific domain. In this paper a technique called HEET(Ħidden WƎb QuƎry Ƭechnique) for modelling and query the hidden web is presented. It proposes a new approach for modelling of consecutive forms and introduce the concept of comprehensive form (consists of many forms).**

***Keywords :*** **Crawler, deep web, wrapper generation, attribute .**

## I. Introduction

Today's web based crawler retrieve content only from a portion of the web called the publicly indexable web (PIW) [1] i.e. the set of web pages which can be seen purely by following hypertext links, by ignoring search forms and the pages which require user authorization or registration. During studies it was observed that significant fraction of web content lies outside the PIW. A large amount of the web and data is hidden behind search forms(large number of data are extracted only through HTML forms).

These types of web are called the hidden web or deep web [2]. Pages in the hidden web are dynamically extracted after submitting the query through different search forms.

The main job of extracting information from the deep web can be categorized as follows :

(1) Formulation of Query or detail description of search method.
(2) Search sources which are relevant to the task.

(3) Fill in search form of source and extract and examine the results of each relevant useful resource.

[1]Assistant Professor, ITM University, Gurgaon
nripendradas@gmail.com

[2] Dean and Associate Professor, Gautam Buddha University, Greater Noida, ela_kumar@gbu.ac.in

It is very much challenging for user to retrieve and analyze relevant as well as important information from the deep web automatically. Now a days, the users manually fills input values to web forms and extract data from the returned web pages. In case of user wish to fill complex queries, filling out forms manually is not feasible, but these queries are required for many web based applications.

In this paper we are concentrating only on one specific domain i.e. Automobile. While browsing different types of automobile web site, it was observed that the web sites are designed in same pattern and users need to fill up number of consecutive forms to retrieve the informations. It is illustrated this with an example given below.

Example 1: As explained above it was observed that all car search related web sites are designed using the same pattern. As shown in figure-1 the first web page consists of the these fields (i) Make field (ii) Radio button with "New car" and "Used cars (iii) Name of state / city / pin code and (iv) Next button. After clicking on next button another new form is opened(called child form figure -2) with more advance search options / text fields e.g. Model, From year, To year, Price Range etc.. After submitting this child form result page is extracted as shown in figure 2.

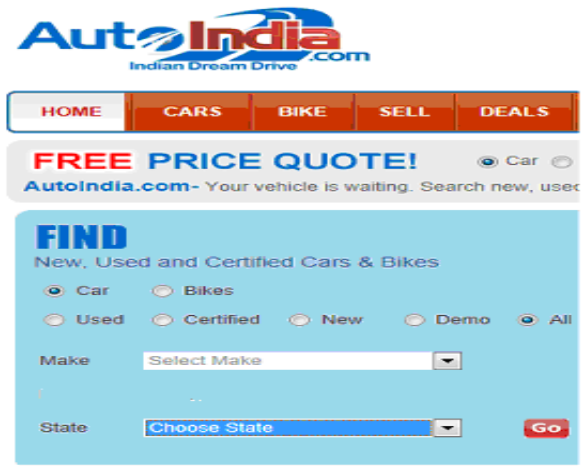


Figure-1 (a) – Existing Search Form

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

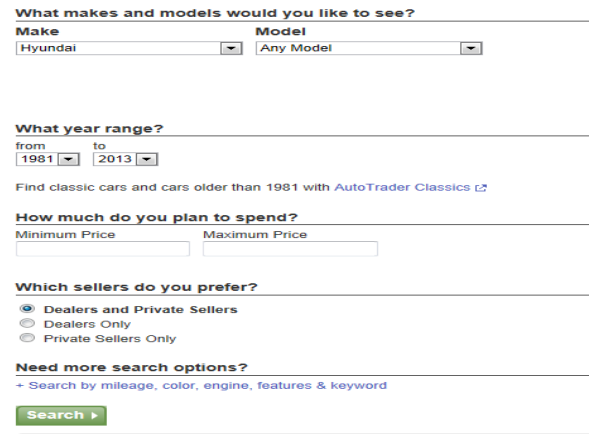


Figure-1 (b) – Child form after submitting the Figure-1(a) form

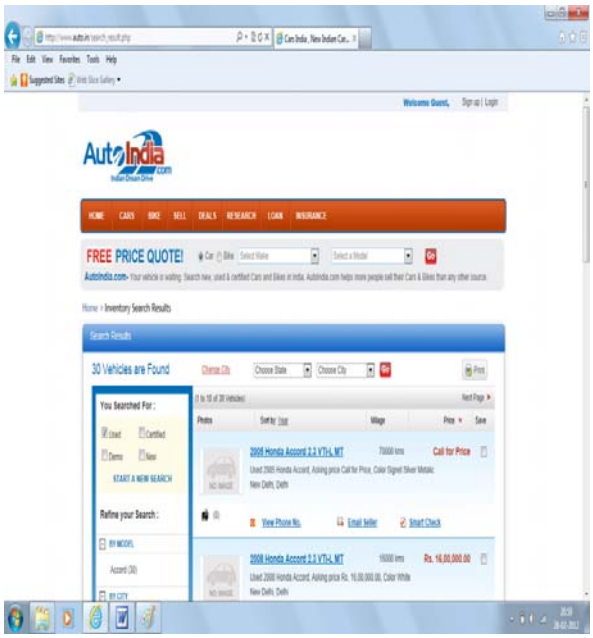


Figure -2 : Result

User may give any type of unexpected query in search engine text box e.g. suppose the user wants to find information about the “used” Japanese red cars of 2006 made within Rs. 15 Lakh and available in “Delhi”. User will have to fillup the form shown in figure 1(a) to formulate such types of query by selecting the choice in make field (e.g. Honda, Toyota etc.). More importantly, User has to fillup and submit the form repeatedly for all different Japanese car. As most of the web pages do not contain colour option, user has to browse each and every result for a particular colour car (Figure-2).

The above process can be efficiently completed by using an automatic form querying system, but it is not an easy task to design this type of automated query processing method due to several challenges.

The challenges are as follows :
(a) Automatic filling of forms : As web pages provides different types of interfaces, automatic filling of forms is a challenging task. Moreover, the user may not be aware of some of the important required field which may by mandatory field for some web site. (e.g. Filling of PIN code to find out the city name is a difficult task for user).

(b) Extraction of results : As most of the data displayed in result pages of web site are embedded in HTML code and this is another difficult problem to extract the result form the web pages. The search and the extraction of required data from such pages are very much complicated task since each web form interface is designed for user’s convenient and every web page format are always different from each other.

(c) Navigational complexity : The pages which are generated after submission of query form may contain link to another web pages consists of relevant informations and therefore, it is necessary to navigate these links to see the detail record. It was also observed that during navigation of such web sites repeated filling of web forms are required which are dynamically generated by the server side programs due to submission of pervious query form. These forms are collectively called consecutive forms.

In this paper, a query technique is discussed for the deep web called HEET (Hidden Web Query Technique) and resolve above mentioned challenges.

The following steps are required for solving the above issues :
(a) If multiple query forms are required to be submitted for extracting the desired results, multiple forms could be normalized to single query form for better and more extraction of data in single submission of query.
The rest of the paper is organized as follows : Section 2 discusses related research in this area. Section 3 and 4 present model for representing normalized forms.

## II. Related Work

The huge growth of the deep web has Motivated interest in the study of web crawlers [3,4]. Currently, many research works are going on for the extraction of information from deep web in better way and many solution have been proposed by the researchers . Some of important types are : (a) Manual Approach (b) Wrapper generation (c) Automatic extraction.

(a) Manual Approach : In this approach programmer tries to extract the information from web pages after identifying it’s schema by writing equivalent source code.

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

(b) Wrapper generation [5] : In this approach, set of rules are designed to extract the information from web pages.

(c) Automatic Extraction[6] : Data items have different meaning and role in web pages. In this approach authors have proposed a method to identify the mapping between data items having same role in different pages.

In this paper, wrapping of data has been considered from different forms to one form to provide the user with more visual and clear presentation of the query results.

## III. Designing of single HTML form

In this section, the attribute-value representation is discussed in a single HTML form. It was observed that the nature and type of the layout markup are not same in HTML forms. The web application must follow the uniformity in designing the form.

### A. *HTML forms.*

An HTML form consists of various parameters as per the requirement e.g. textboxes, dropdown menus, radio buttons, check boxes including banners, advertisement, diagram etc.. After filling up all the required information available in forms user submit the form to web server or mail server for further processing. HTML form are designed with the HTML code and a form is embedded inside the <FORM> tag. Each HTML from contains a set of form field and the URL address of server side program so that whenever user submits the query form, server returns a set of result pages.

To process a FORM in server side mainly three essential attributes are required :
(a) Action attribute – It contains the URL address of the form .
(b) Method attribute – HTTP method is used to submit the form.
(c) Enctype attribute – It specify the content type used for form processing.

### *B. Modeling of consecutive forms*

In the previous section it was discussed how to design a single form. Sometime the desired results were retrieved by using only a single form called root form. But it was observed in some forms(e.g. auto-search form, book-search form etc..) that user need to fillup more than one form known as the child form(s) or descendant form(s) to retrieve the desired result as shown in figure 1(a) & figure 1 (b) because the root form has dependency on child form.

Example : As shown in figure-1, if the user wants to search for different model of Maruti make, in this case user has to select the Maruti make in choice make field and after submitting this root form the second form will display all models of Maruti make as shown in figure 1(b).

### *C. Issues in modelling of consecutive forms*

Some of the issues were observed in modelling consecutive forms. Most consecutive forms have dependencies between the main and descendant form.
Example : In search interface form as shown in figure-1, user requires two consecutive forms to be fill out, if user searched for “Used Car” & “New Car”. It was also observed that form dependency exist between form field. For example, if user selects the “Maruti” in make menu , the all models of “Maruti” make will be displayed in its child form in drop down menu option and if the user wants to see the details of all model one by one he has to select one model at a time and submit the query. It means if there are 10 models available for “Maruti” make(in our example), user has to select 10 times and submit the query to retrieve the results of different model , which is tedious as well as time consuming job.

## IV. Comparison of Existing and New Approach

The comparison of existing and new approach can be described through flow chart :

*A. Existing Method :* Here the example of Automobile domain have been taken and its form.

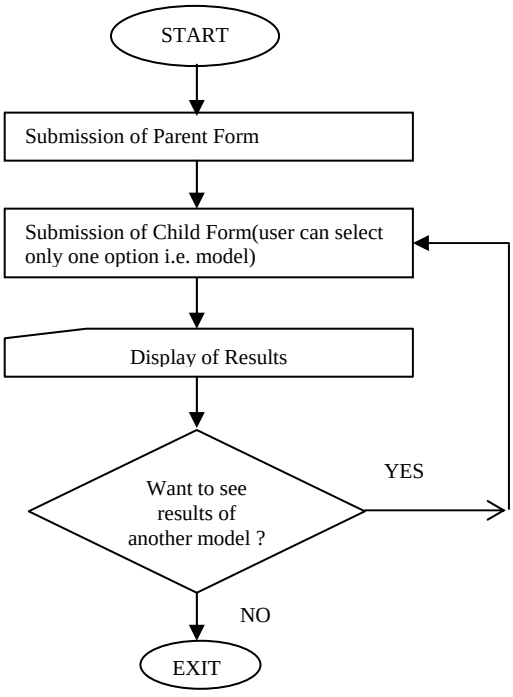


Figure – 3 Flow chart of the existing system

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

*B. New Approach*

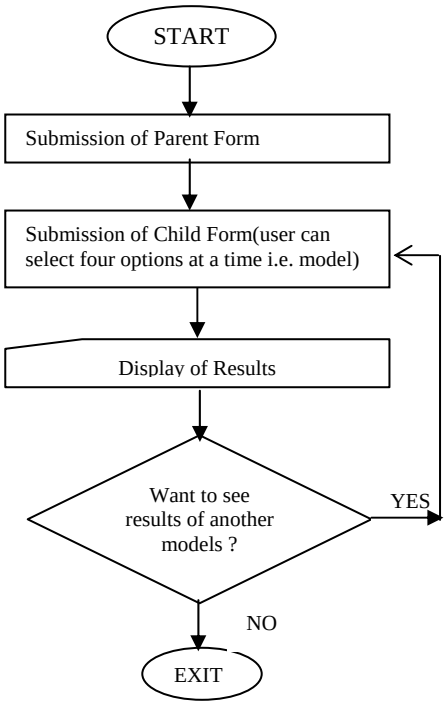


Figure -4 Flow chart of the new proposed system

It was observed that in existing form query interface it allows user to fill up / select only one option from the given option field, in such case if a user wants to see the model of different cars with same options(i.e. price range, city etc..) the user has to first return to the child form once again to fill up the form and submit the query which takes more submission and retrieval time.

Example : In the existing single form query it was observed that the First Form or Parent Form contains the following attributes :
Parent_Form {make, city, search_type, submission_id}
Where make=”maruti”, city=”Delhi”, search_type = ”used” & submission_id = ”76433”.

And when Parent Form is submitted a Second Form or Child Form appears in the screen which contains the following attributes (Figure 1(b) ) :

Child_Form {model, StartYear, EndYear, MinPrice, MaxPrice, SellerType}Where model = “SX4”, StartYear=2005, EndYear=2008, MinPrice=5 Lac, MaxPrice = 7 Lac SellerType=”Dealer” or “Private Seller” or “Owner”.

If a user wants to see 4 different models of same make (e.g. maruti) , then 4 different times user will have to select the model to retrieve the desired results which is a lengthy as well as time consuming process.

But in our suggested model the retrieval time will be very less as compared to the existing one.

Example : it is presumed that user will select 4 make of car at a time and due to this option the corresponding 4 model option will appear in the query form and user can select 4 model option of concerned car make. In this case the submission as well as retrieval time will be less as compared to existing one.

*A. Time required to extract the information*

In earlier system, If $t_1$ time requies to get the child form, after submitting parent form and $t_2$ time requies to extract the informaiton after submitting the child form.
Then total time requies to extract the information for a single make query form is :

Total Extraction time = $(t_1 + t_2)$ seconds. And
When user wants to see 4 different models of same make e.g.
Make=”maruti” (a) model = “Maruti alto”, (b) model = “Wagon R” (c) model = “SX4”
(d) model=”swift dzire”

Total Result Extraction Time(approx) = $4(t_1 + t_2)$ because user can select one model at a time.

But this is not the case with our proposed system. User will have facility to see the result of 4 different model of same make or different make of different model in single submission (as shown in fig-5).

Then,
Approximate Total Result Extraction Time = $t_1$ second.

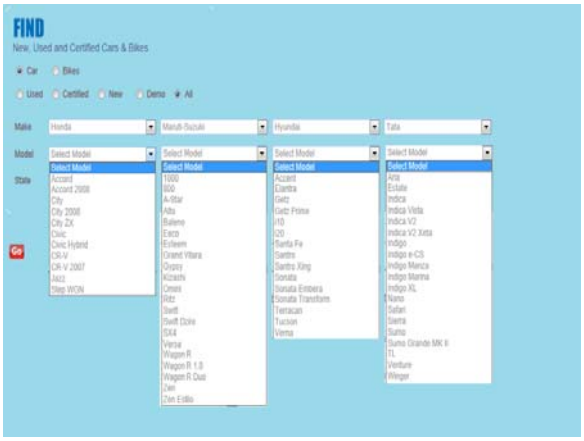


Figure – 5 : Snap Shot of proposed work

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

## VI. Conclusion

In this paper domain dependent approach have been described for retrieving the data behind a given form. In particular , a novel approach have been proposed for modelling the consecutive forms into a single form for more results in a single submission of query form which saves the query submission time, execution time, result extraction time.

## References

[1] S. Lawrence, C. L. Giles, Searching the world wide web, Science 280(5360)(1998) 98-100.

[2] M . K. Bergman , The Deep Web : Surfacing Hidden Value, September 2001, http://www.brightplanet.com/ deepcontent/tutorials/Deepweb/deepwebwhitepaper.pdf.

[3] S. Chakrabarti, M. van den Berg, B. Dom, Focused Crawling : A new approach to topic specific web resource discovery, in 8th world wide web conference may 1999.

[4] J . Cho, H. Gracia-Molina . The evolution of the web and implication for an incremental crawler , in : Proc, 26th Int. Conf. Very Large Data Bases. VLDB, 2000.

[5] Yanhong Zhai and Bing Liu, “Extracting Web Data Using Instances Based Learning.” WISE conference, 2005.

[6] Hu D, Meng X, “ Automatic Data Extraction from Data-Rich Web Pages” , The 10th Data System for Advanced Applications (DASFAA), Beijing, 2005.

[7] LIU Wei, MENG Xiao-Feng, MENG Wei-Yi, A Survey of Deep Web Data Integration, Chinese Journal of Computers, Vol. 30, No. 9, Sept 2007, pp : 1475-1489.

[8] B. He, Patel M, Zhang Z, C Chang, Accessing the deep web, Communication of the ACM, Col. 50 (5), May 2007; 95-101.

[9] Yoo Jung An, James Geller, Yi-Ta Wu, Soon Ae Chun : Semantic deep web : automatic attribute extraction from the deep web data source. In proceeding of the 2007 ACM symposium on Applied Computing (SAC2007), Seoul, Korea, March, 2007, pp : 1667-1672.

[10] Longzhuang Li, Y onghuai Liu, Abel Obregon, and Matt A.Weatherston "Visual Segmentation-Based Data Record Extraction from Web Documents," IEEE IRJ 2007.

[11] Wei Liu , Xiaofeng Meng, and Weiyi Meng, "ViDE: A Vision-based Approach for Deep Web Data Extraction," IEEE TKDE, 2009.

[12] Hongkun Zhao, Weiyi Meng, Zonghuan Wu, Vijay Raghavan, and Clement Yu, "Fully automatic wrapper generation for search engines," in ACM WWW, 2005.

[13] Wu, W., Doan, A., Yu, C.: WebIQ: Learning from the Web to match Deep-Web query interfaces. In: Proceedings of the 22nd IEEE International Conference on Data Engineering (ICDE 2006), p. 44 (2006).

[14] B. He, Z. Zhang, and K. C.-C. Chang. Knocking the door to the deep web: Integrating web query interfaces. In *SIGMOD Conference, System Demonstration*, 2004.

[15] L. Barbosa and J. Freire. Siphoning hidden-web data through keyword-based interfaces. In *SBBD*, 2004.

[16] A. Ntoulas, P. Zerfos, and J. Cho. Downloading hidden web content. Technical report, UCLA, 2004.

[17] J. Cope, N. Craswell, and D. Hawking. Automated Discovery of Search Interfaces on the Web. In *Proc. of ADC*, pages 181–189, 2003.

ISBN: 978-988-19251-6-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)