Hidden Web Data Extraction Using Dynamic Rule Generation

Anuradha
Computer Engg. Department
YMCA University of Sc. & Technology
Faridabad, India
anuangra@yahoo.com

A.K Sharma
Computer Engg. Department
YMCA University of Sc. & Technology
Faridabad, India

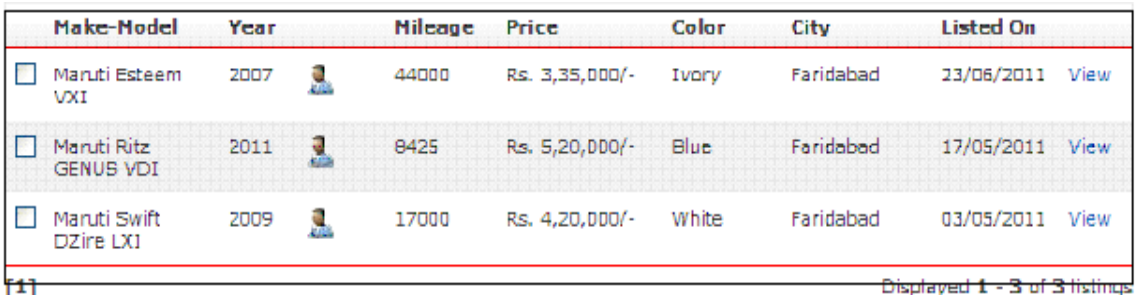
***Abstract*— World Wide Web is a global information medium of interlinked hypertext documents accessed via computers connected to the internet. Most of the users rely on traditional search engines to search the information on the web. These search engines deal with the Surface Web which is a set of Web pages directly accessible through hyperlinks and ignores a large part of the Web called Hidden Web which is hidden to present-day search engines. It lies behind search forms and this part of the web containing an almost endless amount of sources providing high quality information stored in specialized databases can be found in the depths of the WWW. A large amount of this Hidden web is structured i.e Hidden websites contain the information in the form of lists and tables. However visiting dozens of these sites and analyzing the results is very much time consuming task for user. Hence, it is desirable to build a prototype which will minimize user's effort and give him high quality information in integrated form. This paper proposes a novel method that extracts the data records from the lists and tables of various hidden web sites of same domain using dynamic rule generation and forms a repository which is used for later searching. By searching the data from this repository, user will find the desired data at one place. It reduces the user's effort to look at various result pages of different hidden websites.**

Keywords-Hidden Web, Crawler, DOM(Document Object Model), Search Engine.

I. INTRODUCTION

The World-Wide Web contains unstructured data as well as structured data. In case of unstructured documents, there is some structure which is enforced by the URL names and hyperlink graph. But the basic view to user is the plain text itself. On other hand, in case of structured data, the structure is enforced by relational tables. Many of these tables contain both relational-style data and a small schema of labeled and typed columns, making each such table a small structured database [4].

Different product companies sell their products using same product features as the other companies do. Some have similar presentation style and some use different presentation. Figure 2 shows the different presentation style as compared to the webpage shown in figure 1. Figure 2 shows a web page segment containing a multiple rows in second column under description field name in contrast to the figure 1 that shows a page segment containing a data table, where data is displayed in the form of rows and columns. Each data record is shown in the single row. It would be easy to extract the information from these websites if they contain same data representation. But the World Wide Web has been dominated by HTML based on a browsing which is designed for easy reading by a human using a standard Web browser, instead of extraction of information by a program. It is therefore difficult to extract the information by same type of HTML parsing method.



Make-Model	Year	Mileage	Price	Color	City	Listed On	
Maruti Esteem VXI	2007	44000	Rs. 3,35,000/-	Ivory	Faridabad	23/06/2011	View
Maruti Ritz GENUS VDI	2011	8425	Rs. 5,20,000/-	Blue	Faridabad	17/05/2011	View
Maruti Swift DZire LXI	2009	17000	Rs. 4,20,000/-	White	Faridabad	03/05/2011	View

[1] Displayed 1 - 3 of 3 listings

Fig 1 Result page of autonagar.com

Photos	Descriptions	City	Year (Model)	Mileage (KM)
Image Not Available	**Maruti Suzuki WagonR LXi BS III** [Rs. 2,55,000] Dealer : Max Marketing (Malik) Phone : 01124318609 Mobile : 9810288402 view details	Faridabad	2007	10000
Image Not Available	**Hyundai Santro Xing XO** [Rs. 2,65,000] Seller : Mr. Peeyoosh Agrawal Phone : 0124-4504916 Mobile : 9711990583	Faridabad	2006	22000

Figure 2. Result page of carsingh.com

The basic idea is to find the information from various hidden web sources and after some processing, present it as a free web search service. Since, Hidden web is a vast repository of data, we confined our research to car domain. This paper proposes a novel prototype that will handle the above said issues. Before discussing the proposed system there are some important annotations about the hidden web data that should be discussed first.

II. BASIC ANNOTATIONS ABOUT HIDDEN WEB DATA

HTML only defines how the data is to be displayed. But the websites under same domain show some uniformities and non uniformities in displaying their data. After observing several sites, some annotations have been made for these kinds of web pages (see figure 1, figure 2). These are as follows:

A. Similarities:

1. Result pages are generated by specific templates. These templates are filled upon matching the fields with the query which is fired by user in search interface.
2. The data items are displayed in some repeated pattern.
3. A group of data items are always presented in contiguous region of web page. This means that the data items are displayed in the form of rows and columns as shown in figure 1and figure 2.
4. The first row contains the attribute names under which the results are shown.
5. All rows except the first show the resultant data that matches the search query.
6. All rows contain same number of columns.
7. The attribute fields have different names in different sites.
8. Generally price or cost is represented by a number itself or the number followed by the string Rs as shown in figure 1.

B. Dissimilarities:

In addition to some similarities shown by presentation of websites of same domain, they also show some another type of irregularities. They may use the different format, syntax to present similar type of information. For example, Table in the web page is created by <TABLE > tag and rows and columns are created using <TR>, <TD> tags. But this is not the one way to create table. They can be created by using several <SPAN>, <DIV> tags. So, due to heterogeneity and lack of same structure, it is really a challenging task to extract the data from various hidden web sources.

To overcome these problems, the data extraction system is proposed that does not consider tags (i.e whether it is <TABLE> tag or it is <SPAN> tag or any other). It takes into consideration the only position of data in the result page and makes a dynamic rule to extract that data. The architecture of the proposed system is given below in figure 3.

III. PROPOSED SYSTEM

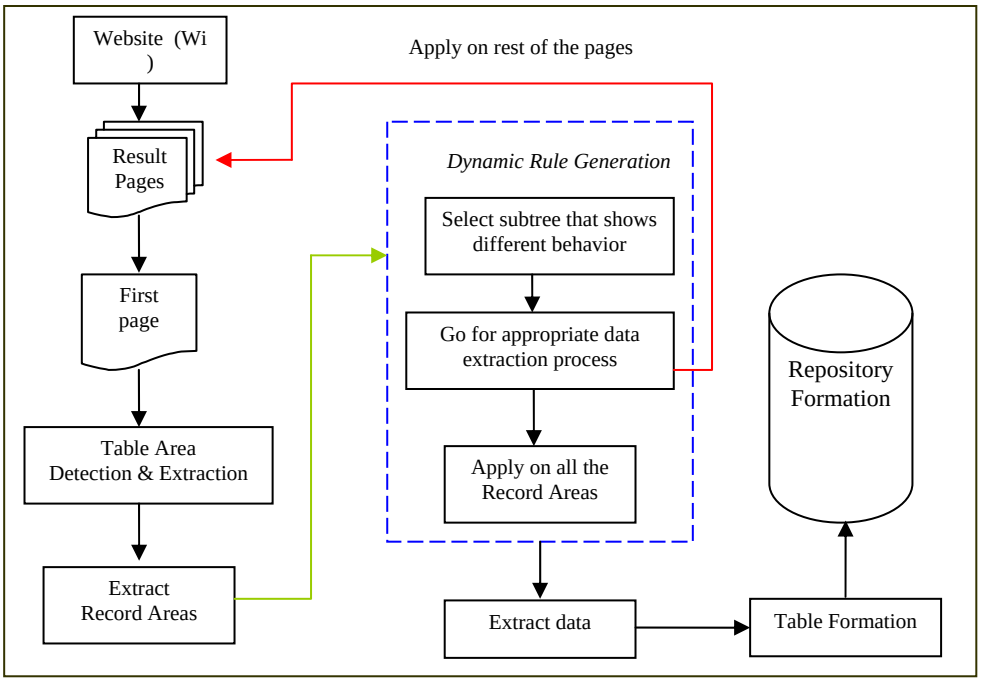


Figure 3. Data Extraction System Architecture

This process works by taking result pages of a particular web site as input. The first page among all the result pages is selected. The result page contains the relevant area and irrelevant area. Relevant area is the area inside which data lies and irrelevant area contains advertisement, navigation link or another type of material that is not a part of result data. So, the next process Table Area detection and extraction is proposed in section A below which detects the relevant area and discard the other ones. It selects the desired table area then this table area is sent for record area extraction (section B). The figure 4 shows the table area, irrelevant area and the data record areas.

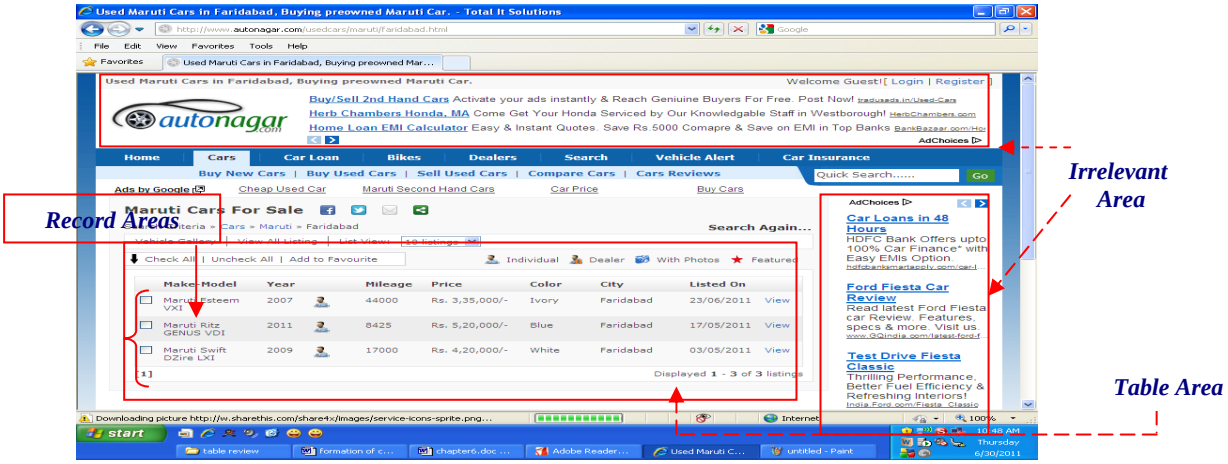


Figure 4. Result page showing all areas

After extracting appropriate table area, next step is to extract the data record areas. These areas are actually the rows of the table. The first row always contain all the field names of table and rest of the rows show the result records under their respective field names. It is seen from figure 1 and figure 2 that number of columns in each row is same. In each column every row shows the same behavior. So, if all the columns of one row show same behavior then this means that data is packed into simple rows and columns of a table. Example of this is result page shown in figure 1 and data records can be easily extracted by simple method i.e data extraction process 1. If

all the columns of one row do not show the same behavior as seen in figure 2, this type of data should be extracted using different method data extraction process 2. So, to analyze the behavior and extract the data, dynamic rule generation technique is proposed in section F. Since all the result pages of a particular website follow the same pattern, data in these pages will also be extracted by the same rule. Therefore for each website, a dynamic rule is created and results are extracted according to it. Now, the next step is to insert this data simultaneously in the table.

A. Table Area Detection and Extraction:

The first step to extract the data from result page is to locate the area of the web page where information lies. The web pages often contain lots of information that is not interesting for the extraction, like advertisement, navigation links, company information etc. as shown in figure 4. So, in order to extract the desired results, we have to omit the other irrelevant area structure from the web page document. [11] defined tables as genuine or non-genuine depending on their data content and structure, and they examine the <TABLE>-tags to find features including layout, content type and word groups in order to classify each table. But this is not sure that tables are only created by < TABLE>-tags. They can be created using <SPAN>, < DIV> etc. So, we cannot detect the table area by looking at <TABLE> tags or <TBODY > tags. Since a HTML document is based on nested tags it can be interpreted as a tag tree or a DOM (Document Object Model) tree [1]. DOM is a standard for accessing and manipulating HTML documents. It presents an HTML document as a tree-structure. In the DOM, everything in an HTML document is a node. The entire document is a document node. Every HTML element is an element node. The text in the HTML elements are text nodes. Every HTML attribute is an attribute node. The root node in the HTML is <html>. All other nodes in the document are contained within <html>. The <html> node has two child nodes; <head> and <body>. The <head> node holds a <title> node. The <body> node holds rest of the document code. Text is always stored in Text Nodes. The value of Text node can be accessed by the innerHTML property in the HTML DOM. Using various DOM functions, we can extract the relevant area and discard the irrelevant one.

B. Relevant Area Extraction:

HTML is the building block of web pages. Most web pages are written in the form of HTML elements that consist of tags enclosed in angle brackets (< >). HTML tags normally come in pairs like <html> and </html>. The first tag in a pair is the start tag, the second tag is the end tag (they are also called opening tags and closing tags). In between these tags web designers can add text, tables, images, etc. Algorithm for Relevant area detection is shown below in figure 5.

```
Algo relevant_areadetect(Wi)
Where Wi is the ith web page from the set of webpages W.)
1. Initialize Stack as an array;
2. strsource = download(Wi);
3. Repeat step 4 to 7 for each tag in strsource until end of file reached.
4. If the tag is a start tag  then
                push (Stack, tag);
   If the tag is a end tag then
                Pop top two elements A, B from the stack
                Where B is the Top element and A is next to Top;
5. string Area = substring( A, B-A);
6. Analyze (Area)                          // Analyze area for domain ontology.
7. If Area is desired area then call data_extract( Area);
   // Desired area is sent for extraction of row and columns.
   Else continue;
```

Figure 5. Algorithm for relevant area detection

This algorithm works by selecting one by one each section from the document and analyze for the desired requirements. Input to this module is resultant webpage. An array is initialized which acts as a stack. Every tag except comment tag (!) is extracted and pushed into the stack as shown in figure 6(b).

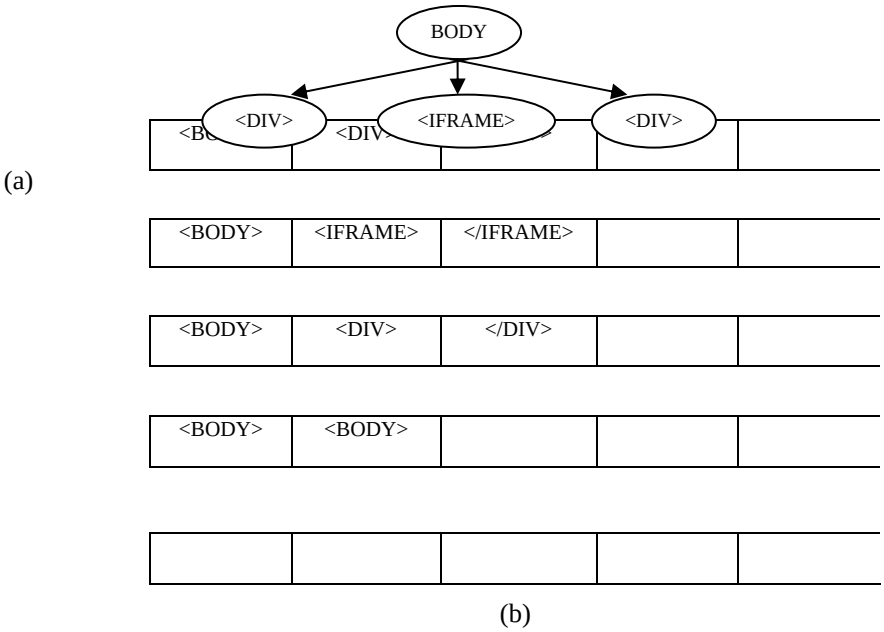


Figure 6. (a) Tree containing three section (b) Stack formation

If the pushed tag is start tag then continue and if it is an end tag corresponding to previous tag then top two tags are popped from the stack and the code between tags is extracted and analyzed until the stack is empty.

C. Domain Attribute Matching:

Since, we are dealing with particular domain (i.e used car) therefore we can find the relevant area by looking at the domain attributes. For example, in used car domain, attributes are make, model, city, price and year. Since, we are getting result pages by submitting the search interface of particular domain. This means the result pages automatically contain domain attributes or their synonyms. So, to analyze every section, Domain attributes should be checked. If the section contains domain attributes, it is selected and others are rejected. For example, algorithm relevant_areadetect( ) is applied on areas shown in figure 6 and only the first section contains domain attributes like make, model city etc. (shown in figure 7). So, this area is relevant area and it is selected for further processing.

```
<DIVclass=advfrm_row><LABEL>Make</LABEL><SPANid=makers><SELECT
style="WIDTH:181px"id=makeclass=selects
onchange="searchmakemodal('http://www.autonagar.com',this.value,'Cars');" name=make>
<LABEL>Model</LABEL> <SPAN id=modelers><SELECT style="WIDTH: 181px" id=modelid class=selcls
name=modelid>><LABEL>Price Range</LABEL> <SPAN id=rangeofprice>
><LABEL>City</LABEL> <SELECT style="WIDTH: 181px" class=selcls name=city>
```

Figure 7. Lines containing domain attributes in first section

Now, the relevant_areadetect( ) algorithm is applied recursively into the depth of first section until it finds the real relevant area.

D. Representation of Data in Hierarchical form:

The nested structure of HTML tags automatically forms a tag tree. This structure reflects the parent-child relationship among the DOM Tree nodes. The DOM tree corresponding to webpage segment (see figure 1) is shown in figure 8. By looking at the DOM tree shown in figure 8 and the basic annotations about hidden web sites discussed in section II. Some conclusions have been made. These are:

1. From the point 6, it is clear that all the sub trees of the DOM tree (all rows of the table) that contain the resultant data have same number of child nodes(number of columns in each row).

2. Point 4 says that the first child node will be inserted as the first row in the table which will be made from this web page and will be displayed as attribute names.

3. Point 5 says that the rest of the child nodes will be inserted as the rows of the table and display the data under respective attribute name.

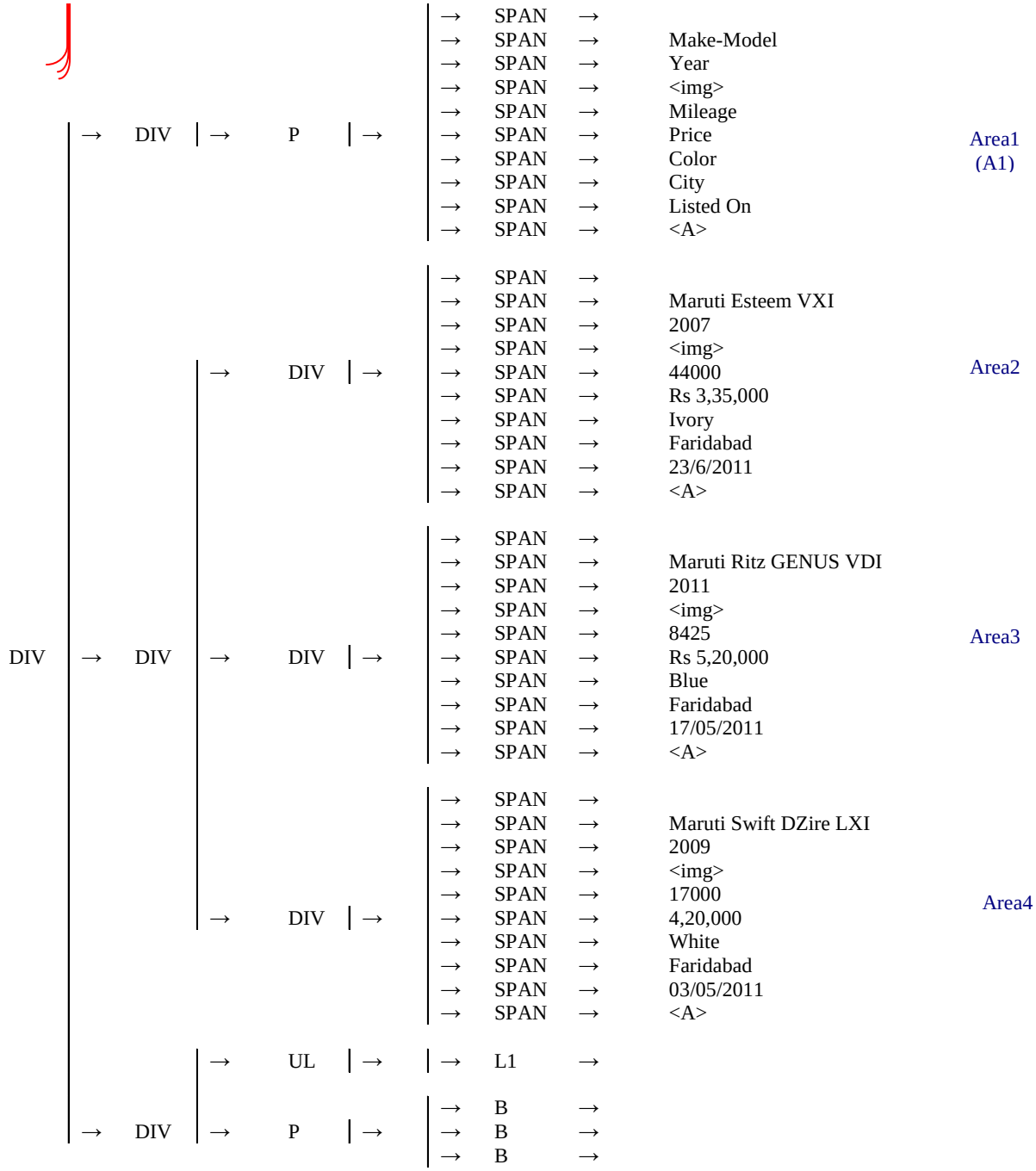


Figure 8. Tree Representation of webpage shown in figure 1

E. Extraction of Record Areas:

To extract the data from the result page, algorithm relevant_areadetect is applied recursively applied to the first section (that is selected in B) and the areas inside that section that contain the domain ontology are extracted. Now, number of childnodes is calculated for each area. The areas with same number of child nodes are selected because from point 6, each row in the table contains same number of columns as shown in figure 1 and these areas are fed as input to dynamic rule generation process.

F. Dynamic Rule Generation:

1) Analyzing the behavior:

As seen in figure 1 and figure 2, the first area contains the field names under which result values are displayed. They always contain atomic values. The different behavior starts from the second row where actual values reside. The reason behind this behavior is that the information is displayed by product company takes human user into the consideration and not the extraction programs. So, number of child nodes of second area is calculated. If it is NULL then all rows are same and show same behavior .They contain data in simple table format as seen in figure 1 and it follows the data extraction process 1 and extract the data. If the second area contains any number of child nodes therefore there are again rows inside this column and all the data records follow the same pattern in each row as seen in figure 2. For this behavior, it will follow the second data extraction process and extracts data according to this process. Algorithm for detecting this behavior is given in figure 9.

```
AAlgo behave-detect ( )

1. Repeat step 2 for each area A2 from the set of extracted areas A;
2. y=calc_childnodes(A2);
3. S= extract first tag of A2;
4. x = document.getElementByTagName(S);
5. m = calc_childnodes(x);
6. if m = = NULL;
   Go for data extraction process with similar behavior.
   Else
   Go for data extraction process with different behavior.
7. End;
```

Figure 9. Algorithm for behavior detection

```
Algo calc.childnodes(Ai )

1. string S= extract first tag;
2. x= document.getElementByTagName(S);
3. z= x.length;
3. return z;
```

Figure 10. Algorithm for calculating children of a node

An algorithm for calculating the child nodes of node is given in figure 10. In this DOM functions are applied. This algorithm works by copying the first tag in string S. The function “document.getElementByTagename( ) returns all the childnodes of node with tag S in x. “x.length” function returns the numner of all the child nodes. This calc.childnodes( ) function returns NULL value if there is no child node linked to node.

The algorithm behave-detect( ) works by looking at this output(number of child nodes). If the node has any child node this means it shows the behavior like figure 2 and if it sees m= Null means all the columns share same behavior then it will extract the data by process 1.

G. Data Extraction:

1) Data Extraction Process 1 and Table Formation:

One way to extract data from HTML pages is to extract all child nodes of each area. New table is created that will consists of first row as the first area where columns are filled by its child node values. The rest of the rows are maintained by rest of the areas and their columns are filled by their respective child node values. From the

figure 8, first row of table created is filled by Area A1 and rest of the rows are filled by A2, A3, A4. For each website, one table is created and filled according to the data packed inside its result page. If we have n websites, then we will have n tables respective to each website.

Table for each website will be created with their respective attributes. When extraction process from all the websites is finished, these tables are merged to form a single repository. The algorithm to extract the data is given in figure 11.

```
Algorithm  data_extract(Wi )

1. Repeat step 2 for each area A1,A2,…,An from the set of extracted areas A;
2. y=calc_childnodes(Ai);
3. Create table T1 with number of columns = y and number of rows = n;
4. k=0;  l=0;
5. Repeat steps 6 to 12 while (l< n)
6. j=0;
7. Repeat step 8 to 11 (while j<y)
8. S= extract first tag of Ai;
9. x = document.getElementByTagName(S);
10. T1[l][k] = x[ j ].innerHTML;
11. j++; k++;
12. l++;
```

Figure 11. Algorithm for data extraction

This algorithm works by taking all the areas as input. These areas will collectively form the table. Number of rows in this table will be number of areas and number of columns will be number of child nodes in each area. So, Each area is selected one by one and the child nodes are calculated for this area by calc_child node( ) function. First area will form the first row of table (columns of which will form the column name of table). The DOM function "document.getElementbyTagname(S) will return all the childnodes in x. The function "x[j]. innerHTML" will extract the text value of all the childnodes. These values are inserted into appropriate cells of the table. This algorithm is illustrated by an example.

Example:

To verify this algorithm, the DOM tree of webpage shown in figure 8 is taken as an example. Area A1, A2, A3, A4 are the areas that contain domain keywords and have same number of children (10).

Number of areas = 4,
Number of child nodes in each area =10
So, Table T1 is created with 10 columns and 4 rows.
N = 4 and y = 10; K=0; l=0;
First tag of First area = <P>
T[0][0] = x[0].innerHTML= " " ,
T[0][1] = x[1].innerHTML =" make-model" ,
T[0][2] = x[2].innerHTML =" year",
T[0][3] = x[3].innerHTML =" Mileage"
T[0][4] = x[4].innerHTML =" Price" and so on...

Similarly, all the child nodes are extracted from tree and inserted into the table. The figure 12 shows webpage segment, its tree representation and the insertion of first area as first row and its child nodes as columns of that row into the table.

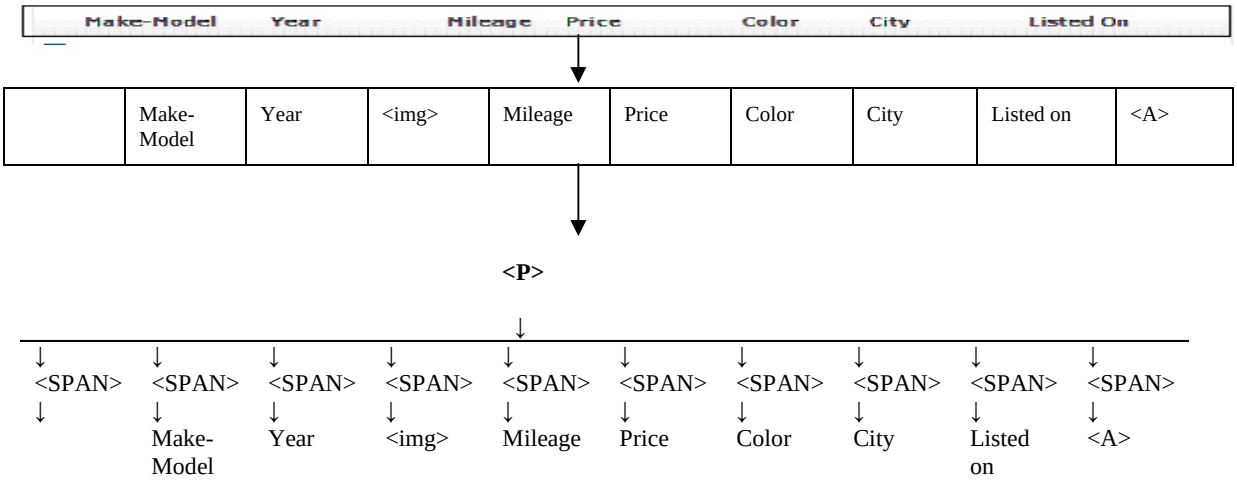
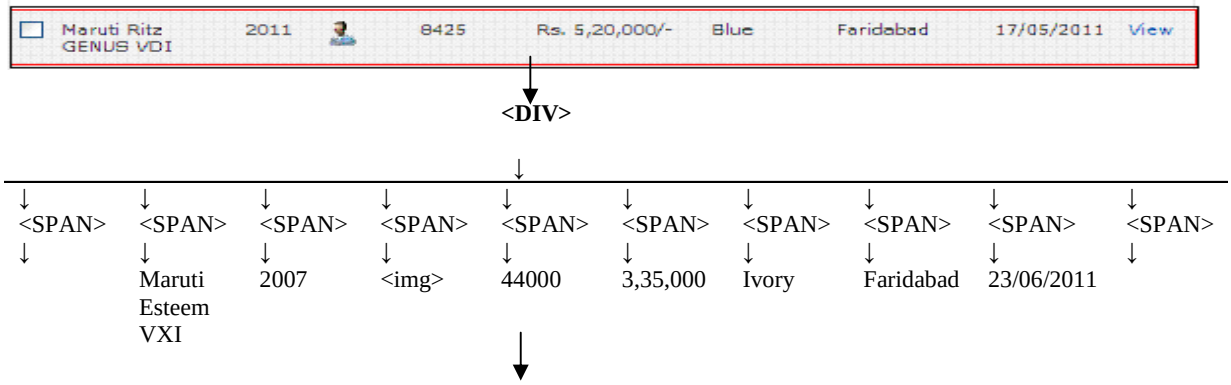


Figure 12. Data extraction process for first area

Figure 13 shows webpage segment, its tree representation and the insertion of second area as second row and its child nodes as columns of that row into the table and this process repeats until all areas are extracted.



	Make-Model	Year	<img>	Mileage	Price	Color	City	Listed on	<A>
	Maruti Esteem VXI	2007	<img>	44000	3,35,000	Ivory	Faridabad	23/06/2011	

Figure 13. Data Extraction for the second area

The algorithm data_extract( ) is applied recursively and extracted data is inserted in the form of rows and columns and form a table at the end as shown in table 1.

Table 1. Table formed after extracting data from all the areas

	Make-Model	Year	<img>	Mileage	Price	Color	City	Listed on	<A>
	Maruti Esteem VXI	2007	<img>	44000	3,35,000	Ivory	Faridabad	23/06/2011	
	Maruti Ritz GENUS VDI	2011	<img>	8425	Rs 5,20,000	Blue	Faridabad	17/05/2011	

	Maruti Swift DZire LXI	2009	<img>	17000	4,20,000	White	Faridabad	03/05/2011	<A>

2) Data Extraction Process 2 and Table Formation:

For the website that shows different behavior from the earlier one as shown in figure 2, the data should be collected and semantically labeled so that they can be appropriately organized into main repository which will be used in later searching. For example, "Maruti Suzuki WagonR Lxi BS III should be inserted into make-model column of main repository and Rs. 2,55,000 should be inserted into price column. Some sites do not provide such labels (price, make-model) when data units are encoded in the returned result page. Like in figure 2, human users can easily tell that the first line indicates make-model and second line tells the price. But the first data extraction system can not predict the labels for the data records. So, meaningful labels should be assigned to these data units.

i) Representation of Data in Hierarchical form:

Tag tree representation of the web page segment (figure 2) is shown below in figure 14.

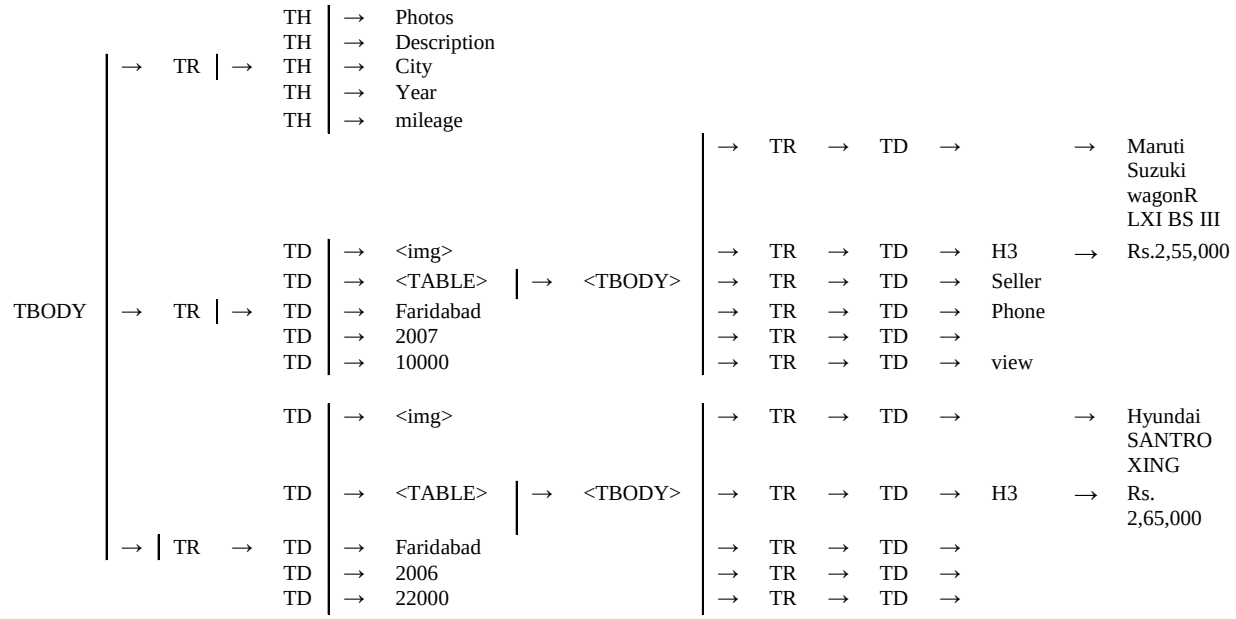


Figure 14.Tree representation of webpage (figure 2)

In this research, Data preprocessing system is proposed which will solve the above problem. The architecture of this system is shown in figure 15.This tag tree shown in figure 14 shows that when we extract the data record by the previous method (A.1), again subtree is encountered inside the second area (second child node) that shows different behavior. So, we send this subtree for preprocessing.

ii) Data Preprocessing:

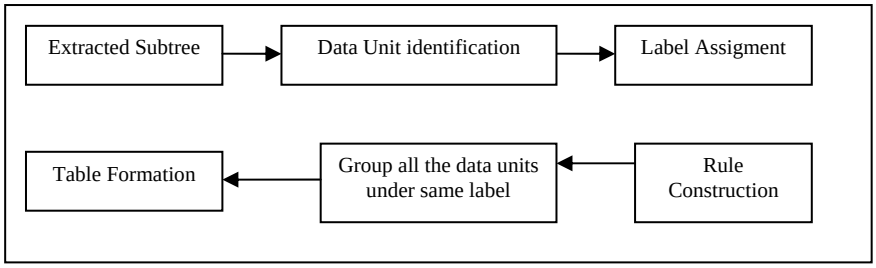


Figure 15. Data Preprocessing System

This system works in five phases. In the first phase, all the data units are identified. The value at the leaf of a tree represents the value in the result page. So, to extract the data unit, the text node values at leaf nodes of this subtree are collected. Now, these data units are sent to label assignment phase. There are some heuristics used to label the data units. These are:

Data Type: Each data unit has its own semantic type. Three basic data types are considered in this approach: Date, Currency and a text string. Each type except text string has certain pattern that can be easily identified. Text string is a normal string. In general, the data units of the same concept have the same data type. For example, price is represented by integer with Rs as prefix. So, the integer values that have prefix come under the field price.

Attribute-value Type: Every search interface has some pre-defined values under each attribute. For example, make of car has a list of values from which user selects a value and submits the form. So, the data units are matched with the values in the Attribute-Value table. Example of this is shown in figure 16 in which Hyundai is value of attribute make and Santro Xing is a value of attribute model. So, "Hyundai Santro Xing combined to come under the group make-model.

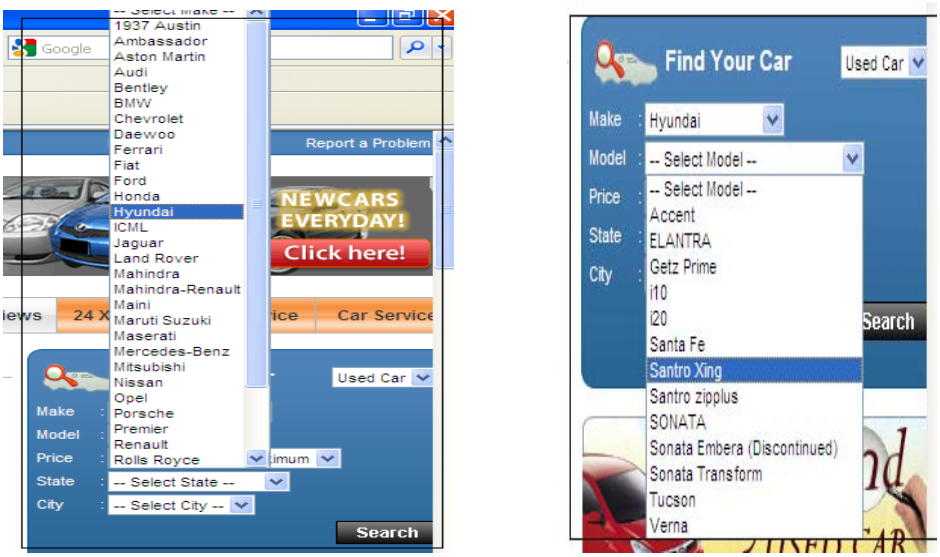


Fig 16. Search interface showing attribute-value pairs

Prefix data matching: A piece of data is sometimes encoded with its label to form a single unit contains both the label and the value but without any obvious separator between them. Such nodes may occur in all rows of the result page. After data alignment, all such nodes would be aligned in one group. For example, in Figure 6.2, label phone contains the phone number. The line containing prefix phone followed by integer values is given below.

<tr><td>Phone : 0124-4504916   Mobile : 9711990583</td></tr>

If the same prefix lies in every row then all the data units are removed from the rows and inserted into the table under phone field.

At the end of this phase, all labels are assigned to the unlabeled data units. Now, at same position of the next row which means at the same position of the child node, data extracted and inserted under the respective field. Now, these new labels are inserted as the column names in the table and values under each label will be stored as row value under the column name. Table shows the new table formed after label assignment phase.

Table 2. Table formed after extracting data from all the areas

Photos	Make-model	Price	Phone	City	Year	mileage

H. Repository Formation:

After extracting rows and columns, separate tables are created for each web site that contains the result records. These tables at the end will be merged into the repository. This repository is used to find the result data corresponding to user's query.

III. CONCLUSION AND FUTURE SCOPE

Hidden Web data integration is a major challenge nowadays. Because of autonomous and heterogeneous nature of hidden web content, traditional search engines have now become an ineffective way to search this kind of data. They can neither integrate the data nor they can query the hidden web sites. Hidden Web data needs syntactic and semantic matching to achieve fully automatic integration. As HTML is a building block for the web documents, this paper proposes a novel method which extracts the individual data units from table data using DOM tree structure of the web page. After collecting data from each table of website, a large repository has been maintained which contains data from various Hidden web sources of same domain and this repository is used for later searching. So, user has to search the data from only one repository and he /she will find the integrated result on one page rather than getting result pages from various websites.

Repository can be formed by merging all the tables of respective websites. But there is a big challenge in merging the tables. The proposed system does not know what regular data records are useful to a user. It simply finds all of them. However, different websites of same domain presents different product options as shown in figures 1 and 2. Some companies name the make as make-model (figure 1) but some denote make as description. Moreover, in a particular application, the user is usually interested in only a specific type of data records. For example, in car domain, user is generally interested in make, model, price, year, mileage, city of the car. So, the system should be designed to output certain important features of product. So, the repository should be formed with these product features only and data the matches these features from different tables should be inserted as rows under respective columns. So, this work can be extended in near future by taking these issues into consideration.

REFERENCES

[1] htttp://www.w3.org/DOM/

[2] B. Liu, R. Grossman, and Y. Zhai, "Mining Data Records in Web Pages," Proc. Int'l Conf. Knowledge Discovery in Databases and Data Mining (KDD), 2003, pp. 601-606.

[3] A. H. F. Laender, B. A. Ribeiro-Neto, A. S. da Silva, and J. S. Teixeira, "A Brief Survey of Web Data Extraction Tools," SIGMOD Record, 2002, vol. 31, no. 2, pp. 84-93.

[4] B. Liu, and Y. Zhai, "NET—A System for Extracting Web Data from Flat and Nested Data Records," Proc. Sixth Int'l Conf. Web Information Systems Eng., 2005, pp. 487-495.

[5] C. H. Chang, and M. R. Gigis, "A Survey of Web Information Extraction Systems", IEEE Transaction on Knowledge and Data Engineering, 2006, Vol.18, No. 10, pp. 1411-1428.

[6] N. Ashish and C. Knoblock. Wrapper generation for semi-structured internet sources. SIGMOD Record, 26(4):8{15, December 1997

[7] N. Kushmerick, D.S. Weld, and R. Doorenbos. Wrapper induction for information extraction. In Proceedings of the 1997 International Joint Conference on Arti_cial Intelligence, pages 729{735, 1997.

[8] I. Muslea, S. Minton, and C. Knoblock. Stakler: Learning extraction rules for semistructured, web-based information sources. In Proceedings of AAAI'98: Workshop on AI and Information Integration, Madison, Wisconsin, July 1998.

[9] 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, Australi,a.

[10] Anuradha, and Sharma, A.K. (2010). A Novel Approach For Automatic Detection and Unification of Web Search Query Interfaces Using Domain Ontology. In International Journal of Information Technology and Knowledge Management, July-December, Vol. 2, No. 2,PP: 196-199.

[11] YalinWang and Jianying Hu. A machine learning based approach for table detection on the web. In WWW '02: Proceedings of the 11th international conference on World Wide Web, pages 242–250, New York, NY, USA, 2002. ACM Press.

[12] Cai, D., Yu, S., Wen, J.-R., and Ma, W.-Y. 2003. VIPS: a Vision-based Page Segmentation Algorithm. Tech. Rep. MSR-TR-2003-79, Microsoft Technical Report.

[13] Simon, K., Lausen, G., and Boley, H. 2006. From HTML documents to web tables and rules. In ICEC, M. S. Fox and B. Spencer, Eds. ACM International Conference Proceeding Series, vol. 156. ACM, 125–131.

[14] Chang, K. C.-C., He, B., Li, C., Patel, M., and Zhang, Z. 2004. Structured databases on the web: observations and implications. SIGMOD Rec. 33, 3, 61–70.

[15] Freitag, D. 1998. Information Extraction from HTML: Application of a General MachineLearning Approach. In AAAI/IAAI. 517–523.

[16] B. Liu and Y. Zhai. NET: System for extracting Web data from °at and nested data records. In *Proceedings of the Conference on Web Information Systems Engineering*, pages 487-495, 2005.

[17] S. Raghavan and H. Garcia-Molina. Crawling the Hidden Web. In Proceedings of VLDB, pages 129–138, 2001.

[18] S. Lawrence and C. L. Giles. Searching the World Wide Web. Science, 280(5360):98–100, 1998.