(IJRST) 2015, Vol. No. 5, Issue No. IV, Oct-Dec **e-ISSN: 2249-0604; p-ISSN: 2454-180X**

WEB DATA EXTRACTION METHOD BASED ON FEATURED TERNARY TREE

***Vidya.V.L, **Aarathy Gandhi**
**PG Scholar, Department of Computer Science,*
Mohandas College of Engineering and Technology, Anad
***Assistant Professor, Department of IT,*
Mohandas College of Engineering and Technology, Anad

ABSTRACT

The Web is a vast and rapidly growing information repository in which data are usually presented using friendly formats, which makes it difficult to extract relevant data from various sources. So web data extractors are used to extract the data from the web pages in order to feed automated processes.web data extraction techniques are usually based on extraction rules that require maintenance if web sources change. In this paper introduced a Featured ternary tree based approach to extract the data from the web pages that share a common pattern, based on this tree generate the regular expression and later it can be used to extract the data from the similar web documents.

Keywords— data extraction, wrapper induction, Data alignment, pattern mining

I. INTRODUCTION

Internet is the biggest information source on the planet .It is difficult to edit the huge amount of data on the web manually. So the concept of web data extraction system was introduced. Web data extraction system is a software system that automatically extracting data from a website. After extracting the data from the web page that extracted data are delivered to a database or some other application. Web data extraction has a wide range of applications such as bioinformatics, analysis of text based documents available to the company, business and competitive intelligence etc. By analyzing the web, we can compare the products, market trends, price details etc.

In this paper introduced a Featured ternary tree based approach, which is the unsupervised method for web data extraction. It works on the one or more web documents at a time and this Featured ternary tree based approach searches for the shared pattern between the web documents and fragments them until finding the relevant data that should be extracted.

After creating the ternary tree, a regular expression is generated from the tree, which represents the template that was used to generate the input documents. Later it can be used to extract the data from the similar web documents.

The remainder of this paper is organized as follows. Section 2 gives a brief overview of Web data extraction related work. Section 3 describes the system overview. Section 4 concludes the paper.

II. RELATED WORK

There are different ways to perform web data extractions. In the earlier stages, manual extraction techniques are used. In that technique, manually writing the programs called wrappers or extractors to extract the data from the web page. Manual extraction technique use some built in rules to extract the data. The working of this technique is based on some prior knowledge of the format of the web page. TSIMMIS, Minerva, Web-OQL, W4F and XWRAP are the examples of manual data extraction [2]. The problem with this technique is that it is a labor intensive task and maintaining wrappers can be expensive and impractical. Therefore, automatic web data extraction techniques are introduced. First supervised techniques are introduced later unsupervised technique are introduced. WIEN, STALKER and SoftMealy are the examples of supervised techniques.

In supervised techniques, wrapper construction system output the extraction rules based on the training examples provided by the designers of the wrapper. The problem with this technique is that designers must manually label the training examples for generating the rules also labelling the training example is time consuming and not efficient enough, so unsupervised techniques are introduced. The advantage of this technique is that no users training examples are needed for web data extraction. IEPAD and OLERA [4] are the some examples of the semi supervised technique. RoadRunner, EXALG, NET, FivaTech are the examples of unsupervised technique.

SoftMealy is the approach to wrapping semi structured web pages. SoftMealy is based on the based on FST (Finite State Transducer) and contextual rules. One of the advantages of SoftMealy is that it handle missing attributes and attributes permutations in the input. The FST consists of two parts such as body transducer and a tuple transducer. The body transducer extract the part of the page that contains the tuples, the tuple transducer iteratively extracts the tuples from the body and it also accepts a tuple and returns its attributes. The main drawbacks of Softmealy are it is not able to generalize overseen separators and it needs many error recovery steps for unseen separators [3].

IEPAD is an information extraction system that automatically identifies the extraction rule by repeated pattern discovery techniques. The repeated patterns are identified using the data structure called PAT trees. PAT tree is a PATRICIA tree (Practical Algorithm to Retrieve Information Coded in Alphanumeric).In previous work, extraction rules are learned from training examples [6]. But in IEPAD, an unsupervised technique is introduced for pattern discovery.

Road Runner is the technique for automatically extracting the data from the HTML sites. In Road Runner, data is extracted through the use of automatically generated wrappers. Road Runner starts with first input page as its initial template. Then match each successive sample pages and checks if the match occurs. If it cannot be, it modifies the current template of the page. Advantages of the Road Runner are it does not require any interaction with the user during the wrapper generation process, it has no prior knowledge about the schema of the web page and also it is not restricted to the flat records, but it can handle nested structures also [5]. The limitations of the Road Runner are number of errors in the input documents affect it's the effectiveness and they don't handle disjunction cases.

EXALG is an algorithm for the extracting the structured data from a collection of web pages generated from the common template. EXALG consist of two stages such as equivalent class generation stage (ECGM) and analysis stage. In ECGM stage, find the sets of tokens having the same frequency of occurrence in every page which are known as equivalence classes [7]. EXALG retains only the equivalence classes that are large and whose tokens occur in a large number of input pages, such type of equivalence classes are known as LFEQs (for Large and Frequently occurring Equivalence classes).The analysis stage constructs the template using the LFEQs. The problem identified in the EXALG is that it is not clear whether EXALG can work on malformed input document or not.

NET is the Nested data Extraction using Tree matching and visual cues. NET is the effective method to extract data from Web pages that contains a set of flat or nested data records automatically. This method is based on a tree edit distance method and visual cues. Advantage of this NET technique is that it enables accurate alignment and extraction of both flat and nested data records. One of the limitations identified in the NET technique is that it incorrectly identifies a flat structure as nested one [8].

FivaTech is a page-level web data extraction technique, which automatically detect the schema of a Website. FivaTech introduce a new structure, called fixed/variant pattern tree. The fixed or variant pattern tree is used for to identify the template and detect the data schema. This technique is the combination several techniques such as alignment, pattern mining Limitations of the FivaTech are searching the longest repeating patterns is time consuming process and also it does not work on the malformed input document without correcting them.[9]

III. SYSTEM OVERVIEW

Over all working of the Featured ternary tree technique is shown in the figure 3.1.

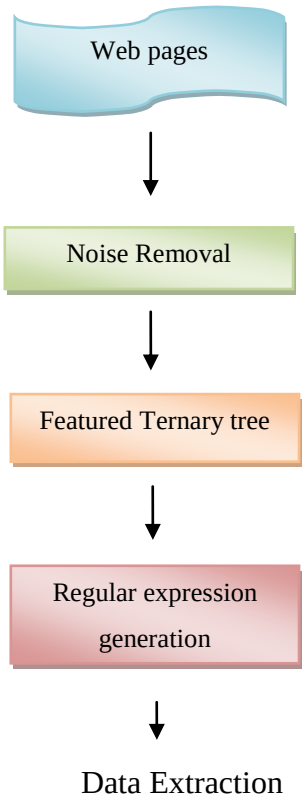

Figure 3.1 General view

Featured Ternary tree based technique consist of three phases. First phase of the Featured ternary tree based technique is noise removal phase. Noise removal phase can be implemented as the preprocessing step for web data extraction. Next phase is the ternary tree creation and regular expression generation. Finally extract the data based on this regular expression. Flow diagram of the noise removal phase is shown in the figure 3.2.

Noise removal phase consist of three steps. In the noise removal phase first create the DOM tree for the HTML page and calculate DOM node weights. Weighting of DOM nodes is the second step in noise removal phase. It can be applied based on the tag importance, position importance and child node importance. Node importance of the parent node is the summation of all the child node importance. Finally normalize the node importance of each node by dividing it with the highest node value in the DOM tree. Noise marking and removing is the final step in the noise removal phase. In noise marking step, compare the node value with the predefined threshold value. If it is lesser than the predefined threshold value, mark it as noisy.

Next step is to remove noisy blocks from DOM tree. For that purpose, a bottom up traversal is done on the tree in such a manner that a parent node is marked as a noisy one if all of its children are noisy. So this marking can be propagated up the tree. Finally the

marked portion of the DOM tree is removed and remaining tree structure is mapped back into HTML page so that a cleaned web page can be obtained.

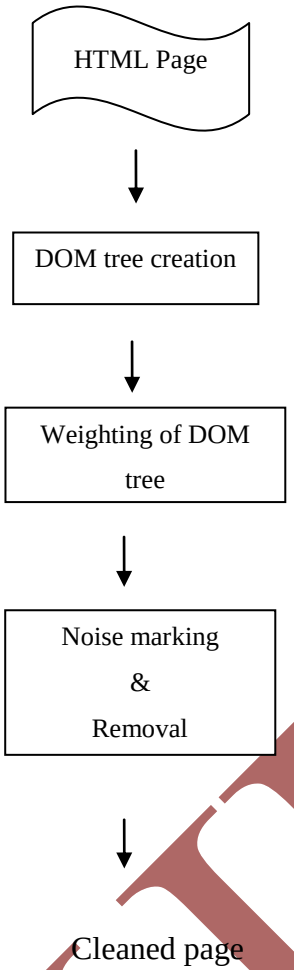

Figure 3.2 Flow diagram of noise removal phase

Next step in the Featured ternary tree based method is the tree creation and regular expression generation. Ternary tree based method takes one or more web documents and set of natural range min and max as input, where min is minimum size of the shared pattern that the algorithm searches and max is the maximum size of the shared pattern. Usually min is set to 1 and max is set to 5% percentage size of the shortest input document specified in terms of tokens. General view is shown in the figure 3.1.

First creates the root node containing all the input documents and set the variable s to max. Starting with this node algorithm loops and searches for the shared pattern of size s, whenever it find the shared pattern between the input documents it creates three new child nodes such as prefix, separator and suffix. These three nodes are organized in the ternary tree. Prefix is the fragment from the beginning of the each text up to the first occurrence of the

shared pattern. Separator is the fragment in between the successive occurrence of the shared pattern. Suffix is the fragment from the last occurrence of the shared pattern to the end of the each text. The process is repeated for all the new child nodes, whenever the shared pattern of size s is not found, variable s decreased .The process is repeated until the s is greater than or equal to the minimum size of the shared pattern. After creating the ternary tree, it is traversed in the pre order and generating the corresponding regular expression that represent the template used for creating the input documents. These regular expressions are later used for extracting the data from the similar documents.

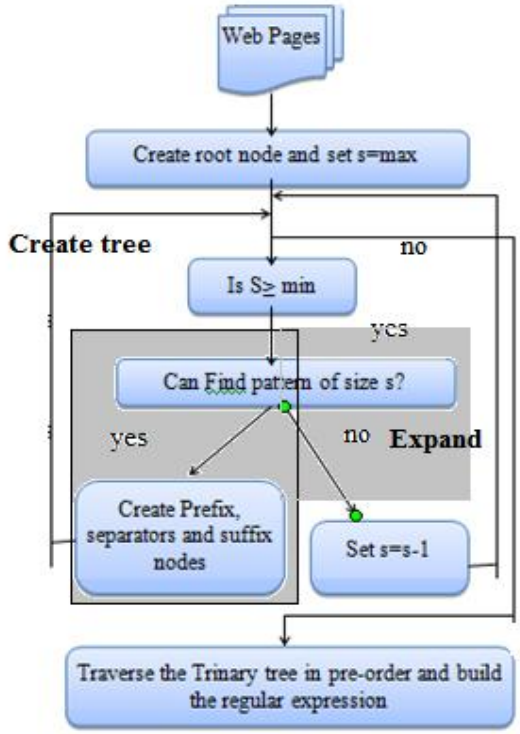

Figure 3.3 General views of tree creation and regular expression generation

IV. EXPERIMENTAL RESULTS

Ternary tree based approach is implemented on a PC with an Intel-Core 1.80-GHz CPU and 4-GB main memory, running Microsoft Windows7 professional. All the algorithms are implemented in Java.

We are performed our experiments on a collection of 104 web documents from 28 datasets,17 datasets were collected from real-world websites and the remaining were downloaded from the RoadRunner and the RISE public repositories. The first group contains the datasets on books, cars, conferences, jobs, movies and real estates. The datasets available at the RoadRunner and RISE repository provide semi-structured web documents

Table 1.performance evaluation based on time

Size of the input document (bytes)		Trinity-Time(msec)	Featured ternary tree-Time(msec)
	174210	5725	4055
	318283	16395	7986
	342278	17690	11246
	1069778	25521	15131

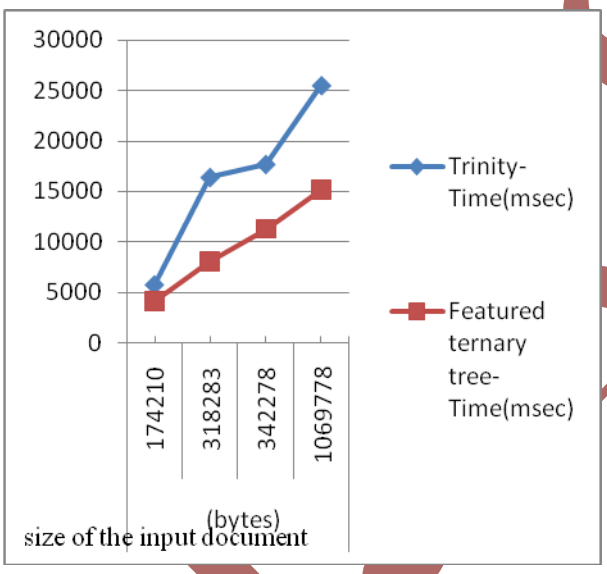

Fig 4.1 performance evaluation based on time

Fig 4.1 and Fig 4.2 shows the performance of the featured ternary tree approach based on memory and the time taken by featured ternary tree. To check the efficiency of our technique, we also implemented the existing algorithm, trinity for comparison. Table 1 and Fig 4.1 show the time taken by the trinity and Featured ternary tree against the size of input documents. Table 2 and Fig 4.3 shows the memory taken by the trinity and featured ternary tree.

By comparing the results Fig 4.1 and Fig 4.2, we found that featured ternary tree is better than the trinity based on time and memory taken.

Table 2.perforamnce evaluation based on memory

Size of the input document (bytes)		Trinity-Memory(no. of tags)	Featured ternary tree-Time(no. of tags)
	174210	23494	15917
	318283	24345	17431
	342278	45671	18056
	1069778	68949	38081

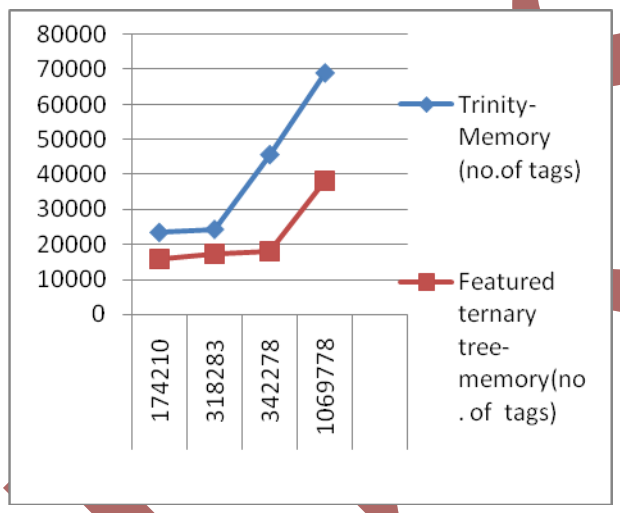

Fig 4.2 performance evaluation based on memory

V. CONCLUSION

Featured ternary tree technique is the efficient method for unsupervised web data extraction. It is based on the principle that web documents which are generated by the same server side template share a common pattern, that do not provide any relevant data and it is the part of the template only. The experimental result shows that the featured ternary tree technique functioned in the best possible manner with the least waste of time and efforts.

REFERENCES

[1] H. A. Sleiman and R. Corchuelo," Trinity: On Using Trinary Trees for unsupervised web data extraction" IEEE Trans.Knowl.DataEng., vol.26, No.6, June 2014.

[2] C.-H. Chang, M. Kayed, M. R. Girgis, and K. F. Shaalan, "A survey of web information extraction systems," IEEE Trans. Knowl. Data Eng., vol. 18, no. 10, pp. 1411–1428, Oct. 2006.

[3] C.-N. Hsu and M.-T. Dung, "Generating finite-state transducers for semi-structured data extraction from the web," Inform. Syst., vol. 23, no. 8, pp. 521–538, Dec. 1998.

[4] C.-H. Chang and S.-C. Kuo, "OLERA: Semi supervised web-data extraction with visual support," IEEE Intell. Syst., vol. 19, no. 6, pp. 56–64, Nov./Dec. 2004.

[5] V. Crescenzi, G. Mecca, and P. Merialdo, "Road runner: Towards automatic data extraction from large web sites," in Proc. 27th Int. Conf. VLDB, Rome, Italy, 2001, pp. 109–118.

[6] C.-H.Chang and S.-C. Lui, "IEPAD: Information extraction based on pattern discovery," in Proc. 10th Int. Conf. WWW, Hong Kong, China, 2001, pp. 681–688.

[7] A. Arasu and H. Garcia-Molina, "Extracting structured data from web pages," in Proc. 2003 ACM SIGMOD, San Diego, CA, USA, pp. 337–348.

[8] B. Liu and Y. Zhai, "NET: A system for extracting web data from flat and nested data records," in Proc. 6th Int. Conf. WISE, New York, NY, USA, 2005, pp. 487–495.

[9] M. Kayed and C.-H. Chang, "FiVaTech: Page-level web data extraction from template pages," IEEE Trans. Knowl. Data Eng., vol. 22, no. 2, pp. 249–263, Feb. 2010.

[10]J. Wang and F. Lochovsky. "Wrapper Induction based on nested pattern discovery." , Technical Report HKUSTCS-27-02, Dept. of Computer Science, Hong Kong U. of Science and Technology, 2002.

[11] Tai, K. The tree-to-tree correction problem. J. ACM, 26(3):422–433, 1979

[12] D. Freitag, "Information extraction from HTML: Application of a general machine learning approach," in Proc. 15th Nat/10th Conf.AAAI/IAAI, Menlo Park, CA, USA, 1998, pp. 517–523.