International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-2 Issue-3, July 2013

Web Data Extraction Using Tree Structure Algorithms – A Comparison

Seema Kolkur, K.Jayamalini

Abstract— Nowadays, Web pages provide a large amount of structured data, which is required by many advanced applications. This data can be searched through their Web query interfaces. The retrieved information is also called 'deep or hidden data'. The deep data is enwrapped in Web pages in the form of data records. These special Web pages are generated dynamically and presented to users in the form of HTML documents along with other content. These webpages can be a virtual gold mine of information for business, if mined effectively. Web Data Extraction systems or web wrappers are software applications for the purpose of extracting information from Web sources like Web pages. A Web Data Extraction system usually interacts with a Web source and extracts data stored in it. The extracted data is converted into the most convenient structured format and stored for further usage.

This paper deals with the development of such a wrapper, which takes search engine result pages as input and converts them into structured format. Secondly, this paper proposes a new algorithm called Improved Tree Matching algorithm, which in turn, is based on the efficient Simple Tree Matching (STM) algorithm. Towards the end of this work, there is given a comparison with existing works. Experimental results show that this approach can extract web data with lower complexity compared to other existing approaches.

Index Terms—About Web Data Extraction, Document Object Model (DOM), Improved Tree Matching algorithm.

I. INTRODUCTION

Web Data Extraction systems [1] are software applications for the purpose of extracting information from Web sources like Web pages. A Web Data Extraction system usually interacts with a Web source and extracts data stored in it and converts the extracted data in the most convenient structured format and stores it for further usage. World Wide Web (WWW), as the largest database, often contains various data that we would like to consume for our needs. This data can be searched through Web query interfaces. The retrieved information (query results) is also called as deep data or hidden data. This information is, in most cases, mixed together with formatting code and other information like website title, advertisement and navigation links, headers, footers, scripts and comments - which makes the page more human-friendly, but not machine-friendly. This deep data is enwrapped in webpages in the form of data records. These special Web pages, which contains these data records are generated dynamically and presented to users in the form of HTML documents along with other content. These webpages can be a virtual gold mine of information for business, if mined effectively. Web Wrapper [2] is a program that extracts content of HTML pages, and translates it into a relational form. In literature, there can be many approaches used for web data extraction. Currently, data extraction methods [3] can be generally divided into following five categories.

- Based on natural language processing – in this method of data extraction, first the structures of clauses and phrases, and the relationship of clauses are analyzed, and then rules for extraction based on the syntax and semantics are generated. Hence, this method is applicable to the source documents which contain a lot of text, especially grammatical text. But the texts in Web pages are usually imperfectly structured sentences, which narrows its applicability. The classical systems that are based on this principle are RAPIER [4], WHISK [5] and so on.
- Based on wrapper summing up the rules – this method of information extraction makes use of machine learning techniques to learn structural features from a number of Web pages, and then sums up the extraction rules using the structural features. Usually, one wrapper can only handle a specific source. To extract information from different sources, a series of wrapper libraries are needed, which requires a huge workload. Tools using this method are mainly WIEN [6] and SoftMealy [7].
- Based on Ontology - The main idea in this method of data extraction is to construct a complete knowledge base for a specific domain. The information is extracted by using the relevant rules in the Knowledge Base to process the Web pages. This method depends less on the structure of the Web page, and requires powerful domain-specific Ontology, which also requires heavy workload. In [8], Shanmugasundaram et al. studied about this method for information extraction.
- Based on HTML structure - this method extracts information based on structural features of HTML documents. Such systems include XWRAP [9], RoadRunner [10] and W4F [11].
- Based on visual features - Visual based wrappers such as ViDE [12], VSDR, and ViPER use visual cue to locate and extract correct data region. These wrappers calculate the boundary and location of a data region, and take data region which is large and centrally located as the correct data region.

In recent years, some value-added services like comparative shopping and domain search have motivated the research of data extracting technology. This paper deals with a Wrapper development, which takes search engine result pages as input, pre-processes these pages, identifies the correct data region, constructs DOM tree, generates template, checks the similarity between template and other webpages using Tree Matching algorithms [3],[14] and converts the data into structured format and stores it. XPath [13] is a language that is used to address parts of an XML documents. It is also used to find and locate the information that the users are interested in structured documents.

Revised Manuscript Received on 30 July 2013.
* Correspondence Author
Ms. Seema Kolkur, Assistant Professor, Computer Department, Thadomal Shahani College of Engineering, Mumbai, India.
Ms. K.Jayamalini,, Assistant Professor, Computer Department, L.R.Tiwari College of Engineering, Mumbai, India.

In this paper, a new approach of data extraction is proposed based on Improved Tree Matching (ITM) Algorithm, which is based on Simple Tree Matching algorithm. For this work, search engine result pages are collected from the website 'books.google.com'. A sample search engine result page which contains data records to be extracted is shown in Figure 1. In Figure 1, the data region is shown in a rectangular box and the data record is shown in rounded rectangle box.

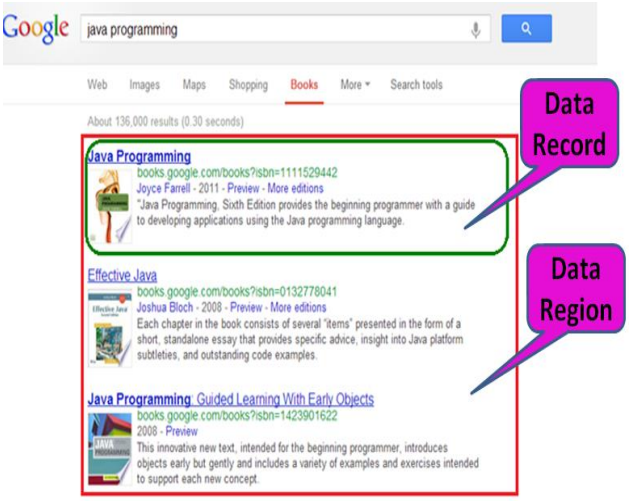

Fig.1. Search Engine Result page from books.google.com

In our work, pages and templates are represented in the form of DOM tree. We would test our system using a set of web pages from books.google.com site. Three different types of tree matching algorithms are used in this work. The results are compared with previous work to showcase that our system is accurate, efficient, and works with reduced complexity.

The rest of this paper is organized as follows:

Section 2 explains system overview, section 3 presents Web data extraction method based on Improved Tree Matching and Section 4 describes the experiment results. Sections 5 and 6 describe result analysis and conclusions. Finally, Section 7 gives an idea of our future work.

II. SYSTEM OVERVIEW

The Block Diagram of the proposed Work is given below in Figure 2. In our system, the search engine result pages are used as input. These pages are first preprocessed and the correct data region is identified. The template pages are randomly selected and the DOM tree is constructed for template pages. DOM Trees for candidate pages are also constructed and the similarity values between pages are calculated using Tree Matching Algorithms.

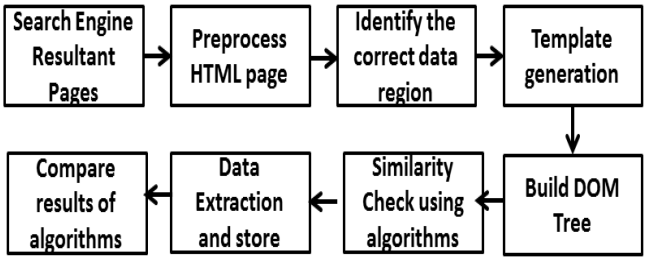

Fig.2. Block Diagram of the proposed Work

Similarity value of two pages which is greater than or equal to threshold value is considered as similar page and data can be extracted from that page and stored in the database in a structured format for future use.

III. SIMILARITY BASED ON IMPROVED TREE MATCHING

In [3], Hua Wang and Yang Zhang, proposed a Simple Tree Matching Algorithm, and in [14], Haikun Hong, Xiaoxin Chen, Guoshi Wu and Jing Li, proposed Extended Tree Matching Algorithm - both make use of dynamic programming to calculate the maximum number of node-pair between two trees. These algorithms recursively find the maximum matching between the first-level subtrees of the two trees by comparing the subtrees of Tree 1 with all the subtrees of Tree 2. The search engine result pages follow similar structure because they are generated automatically by a pre-defined template. In this paper, we use the Improved Tree Matching Algorithm to calculate the similarity between two HTML documents. Experimental results show that this method can perfectly reflect the similarity of two documents with reduced complexity when compared with STM.

Let $A = R_A :<A_1, \ldots A_m >$ and $B = R_B :<B_1, \ldots Bn >$ be two trees, where R_A and R_B are the root of A and B respectively; A_i and B_j be the ith and jth node of the first-level subtrees of A and B, respectively. If the symbols R_A and R_B are the same, then the maximum matching of A and B is calculated by comparing the nodes in subtree 1 of tree A with nodes in subtree 1 of tree B. If there is match M(A,B) will be incremented by 1. Then nodes in subtree 2 of Tree A are compared with nodes in subtree 2 of Tree B. This process will continue for all the nodes in all the subtrees of tree A and tree B. If R_A and R_B contain distinct symbols, then M $(A,B) = 0$.

The Improved Tree matching algorithm is explained below.

Algorithm 1: Improved_Tree_matching algorithm

1. if the roots of the two trees A and B contain distinct symbols then
2. return 0;
3. Else
4. m = Size of A;
5. n = Size of B;
6. Initialization: M[i, 0] = 0 for i = 0, … , m;
7. M[0, j] = 0 for j = 0, …, n;
8. for i = 1 to m do
9. for j = 1 to n do
10. if(A[i][j]==B[i][j])
11. match++;
12. M[i,j]=match;
13. endfor
14. endfor
15. return maximum(M);
16. endif

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

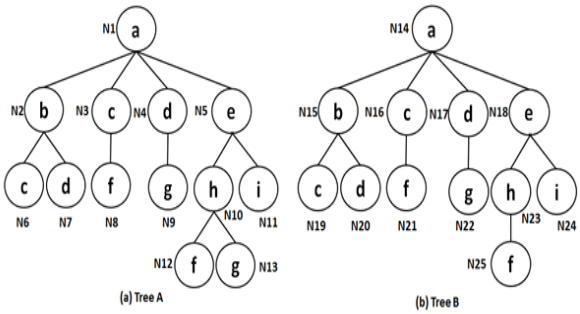

Fig. 3 Two Sample DOM Trees

To better understand the algorithm and to implement, let us take two trees shown in Figures 3(a) and 3(b). First, the roots of two trees are compared. If they contain the same symbols, the algorithm finds the maximum matching between the first-level subtrees of the two trees, and saves the matching in the M matrix. If they contain distinct symbols, the two trees totally do not match. First, the roots, nodes N1 and N14 are compared. Since they contain identical symbol, the maximum matching of the ith and jth first-level subtrees of A and B is calculated and stored in M matrix. In the above example, N3 of tree A is compared with N15 of tree B. The matching node-pairs between first subtree of both the trees is equal to 3, which is stored in M matrix. The construction of M matrix for the above example is shown in Figure 4.

	N15	N16	N17	N18
N3	3	-	-	-
N4	-	5	-	-
N5	-	-	7	-
N6	-	-	-	11

Fig 4 .M matrix for the first-level subtrees

The number of matching nodes pairs between first level subtrees of Tree A and Tree B of figure 2 is 11.

IV. PREPROCESSING AND DATA EXTRACTION

Our data extraction can be divided into seven parts: HTML document pre-processing; Identifying Correct Data Region; Construction of DOM Tree; Compression of DOM Tree; Template Generation; Similarity Check using Tree Matching Algorithm; Storing and Extracting Data.

a. HTML Document Pre-Processing

Html found on Web is usually dirty, ill-formed and unsuitable for further processing. The semi-structured features of HTML language make the HTML source files non-standardized. For any serious consumption of these documents, it is necessary to first clean them up. There are several mature html parsers like HtmlCleaner, NekoHTML, Html Parser, Jsoup and JTidy that can do this job. For this work we've chosen the HtmlCleaner for use.

b. Identifying Correct Data Region

HtmlCleaner is an open-source html parser written in Java which can reorder individual elements of a Html document and produce well-formed XML documents. The pre-processed web pages are converted into structured hierarchical tree (XML) using HtmlCleaner. It also provides methods for tag filtering, which are used to identify and extract the correct data region.

c. Construction of DOM Tree

The DOM views HTML and XML documents as a tree-structure. All elements of these documents can be accessed through the DOM tree. The elements, their text, and their attributes are all known as nodes. After identifying data region, DOM tree is constructed for that specific region. DOM Tree is constructed using Apache's xercesImpl.jar file. The DOM tree for sample page 1 is shown in Figure 5.

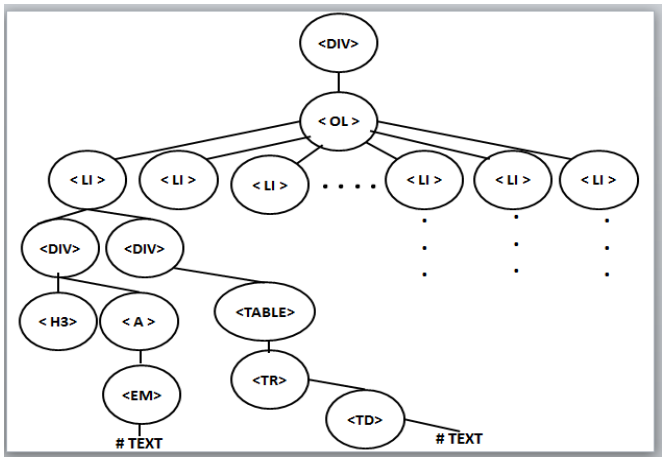

Fig. 5.DOM tree of Sample page 1

d. Compression of DOM Tree

Nowadays, many formatting tags like <font>, <b>, <i>, , <heading>, <table> etc. are used in the design of webpages to make these pages human friendly. These tags increase the size of the DOM tree. Hence, to ensure the size of the DOM tree is optimal, the unwanted tags are removed.

e. Template Generation

Dynamically generated pages are usually generated automatically by the pre-defined templates, and the templates handle data of the same source with a fixed display format. Therefore, they have very similar page structure. The approach here would be to enter a query through a Web browsing interface, followed by obtaining the code segment corresponding to the record from the result page, and convert it to a DOM tree. DOM tree is the matching template tree, T, for the extraction of rules.

f. Similarity Check

Tree Matching Algorithms are used for similarity check. Let's write A and B for DOM trees corresponding to the two HTML documents, respectively. Then we can define the similarity of A and B (formula 1).

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

$$Similarity(A,B) = \frac{ImprovedTreeMatching(A,B)}{(Size(A)+Size(B))/2} \quad \ldots (1)$$

Where ImprovedTreeMatching (A, B) represents the number of maximum matching nodes of tree A and tree B; sizes(X) represents the number of nodes on tree X. When Similarity(A,B) is closer to 1, Tree A and Tree B are very similar to each other, and the HTML documents they represent are also very similar and the Web data will be extracted from that page ; otherwise, the two trees are considered as not similar. In this paper, the threshold value is set as 0.6.

g. Comparison of Results

At the end of this work, the results of Simple Tree Matching (STM) algorithm, Extended Simple Tree Matching algorithm (ESTM) and Improved Tree Matching (ITM) algorithm are compared.

V.RESULTS AND ANALYSIS

Many sample search engine resultant pages are taken and compared with ten different template pages using all three algorithms. The similarity values for five sample pages (p1, p2, p3, p4 and p5) with 10 templates using STM is shown below in Figures 6(a) and 6(b).

Pages ↓ / Templates →	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
p1	1.28	1.25	1.27	1.28	0.89	0.88	0.89	1.28	1.27	1.25
p2	1.25	1.21	1.23	1.25	1.02	1	1.02	1.25	1.23	1.21
p3	1.27	1.23	1.25	1.27	1.01	0.99	1.01	1.27	1.25	1.23
p4	1.28	1.25	1.27	1.28	0.89	0.88	0.89	1.28	1.27	1.25
p5	0.89	1.02	1.01	0.89	1.2	1.18	1.2	0.89	1.01	1.02

Fig. 6(a) Similarity values of pages using STM

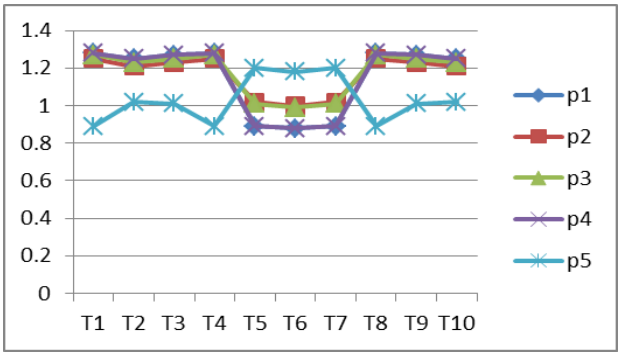

Fig. 6(b) Output of STM

The similarity values for five sample pages (p1, p2, p3, p4 and p5) with 10 templates using ESTM is shown below in Figures 7(a) and 7(b).

Pages ↓ / Templates →	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
p1	0.72	0.68	0.7	0.72	0.61	0.6	0.61	0.72	0.7	0.68
p2	0.68	0.71	0.69	0.68	0.63	0.62	0.63	0.68	0.67	0.68
p3	0.7	0.69	0.71	0.7	0.62	0.62	0.62	0.7	0.68	0.69
p4	0.72	0.68	0.7	0.72	0.61	0.6	0.61	0.72	0.7	0.68
p5	0.61	0.63	0.62	0.61	0.67	0.65	0.67	0.61	0.62	0.63

Fig. 7 (a) Similarity values of pages using ESTM

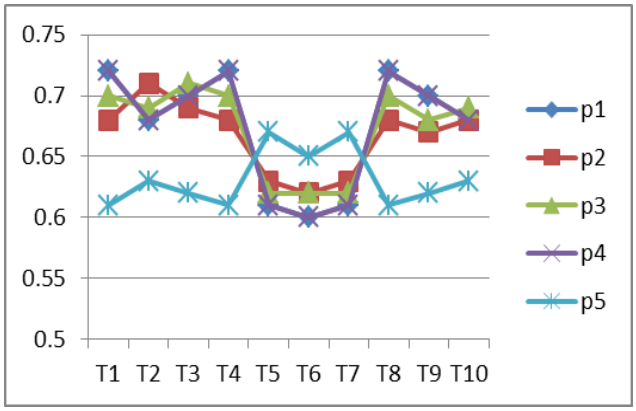

Fig. 7 (b) Output of ESTM

The similarity values for five sample pages (p1, p2, p3, p4 and p5) with 10 templates using ITM are shown below in Figures 8(a) and 8(b).

Pages ↓ / Templates →	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
p1	1	0.92	0.96	1	0.75	0.75	0.75	1	0.96	0.92
p2	0.92	1	0.96	0.92	0.83	0.79	0.83	0.92	0.88	0.92
p3	0.96	0.96	1	0.96	0.79	0.79	0.79	0.96	0.92	0.96
p4	1	0.92	0.96	1	0.75	0.75	0.75	1	0.96	0.92
p5	0.75	0.83	0.79	0.75	1	0.93	1	0.75	0.79	0.83
p6	0.75	0.79	0.79	0.75	0.93	1	0.93	0.75	0.76	0.83

Fig. 8(a) Similarity values of pages using ITM

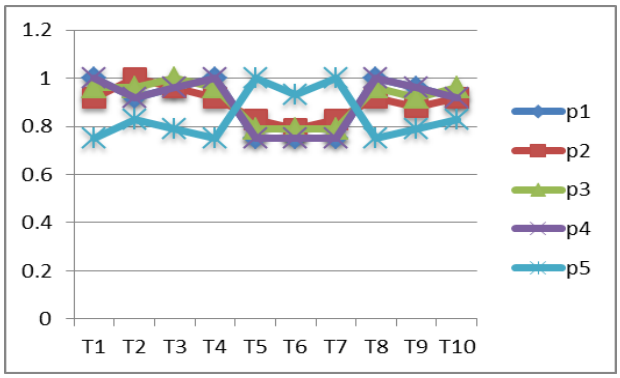

Fig. 8 (b) Output of ITM

All three algorithms give the same result i.e. similarity value greater than the threshold value (0.6) on search engine result pages but the proposed Improved Tree Matching algorithm works in reduced complexity. The efficiency of ITM is shown in terms of execution time and number of loops used - illustrated below in Figures 9(a) and 9(b).

Algorithm	Number of loops used	Execution Time
STM	3	1469
ESTM	3	1250
ITM	2	1050

Fig. 9 (a) Comparison of Algorithms

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

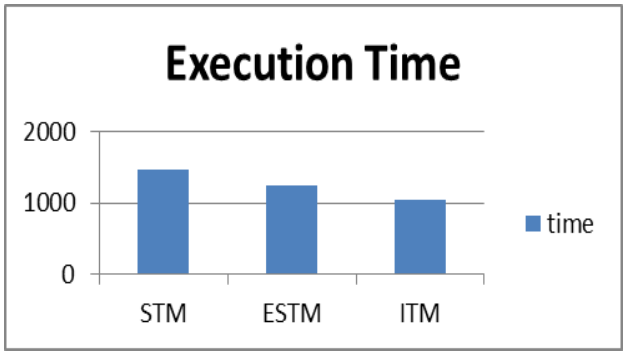

Fig. 9 (b) Graph of Comparison of Algorithms

VI. CONCLUSION

To extract data records from similar Web pages effectively, in this paper, we propose a Web data extraction method based on Improved Tree Matching algorithm. The proposed method takes full advantage of the structural similarity of the Web pages generated by the templates. Experimental results on sample Webpages show that the proposed web data extraction method can effectively extract data records from similar Web pages, with reduced complexity. The proposed method is shown to be a very effective web data extraction method.

VII. FUTURE WORK

In future, we plan to study extracting data from Webpages which have different in structures. To meet people's demand of intelligent information, we will also study how to reduce the manual involvement, and establish intelligent information retrieval systems.

ACKNOWLEDGMENTS

We wish to express our sincere thanks and deep sense of gratitude to all the staff members of computer department of Thadomal Shahani College of Engineering, Mumbai for thier support. We also wish to thank to our family members for their support to complete this work.

REFERENCES

1. http://arxiv.org/pdf/1207.0246.pdf (Accessed: May, 2013).
2. C. H. Chang, M. Kayed, M. R. Girgis, and K. Shaalan, "A survey of web information extraction systems," IEEE Transactions on Knowledge and Data Engineering, 2006, pp. 1411-1428.
3. Hua Wang, Yang Zhang, "Web Data Extraction Based on Simple Tree Matching,", IEEE, 2010, pp. 15-18.
4. M. E. Califf, and R. J. Mooney, "Relational learning of pattern-match rules for information extraction," Proceedings of the ACL Workshop on Natural Language Learning, Spain, July 1997, pp. 9-15.
5. N. Kushmerick, D. S. Weld, and R. Doorenbos, "Wrapper induction for information extraction," Proceedings of the 15th International Conference on Artificial Intelligence (IJCAI), pp. 729-735, 1997.
6. N. Kushmerick, D. S. Weld, and R. Doorenbos, "Wrapper induction for information extraction," Proceedings of the 15th International Conference on Artificial Intelligence (IJCAI), pp. 729-735, 1997.
7. C. N. Hsu, and M. T. Dung, "Generating finite-state transducers for semi-structured data extraction from the web," Journal of Information Systems, vol. 23(8), pp. 521-538, 1998.
8. J. Shanmugasundaram, J. Kiernan, E. Shekita, F. Catalina, and F.John, "Querying XML views of relational data," Proceedings of the 27th VLDB Conference, Rome, Italy, 2001, pp. 261-270.
9. L. Liu, C. Pu, and W. Han, "XWRAP: An XML-Enabled Wrapper Construction System for Web Information Sources," Proceedings of the 16th International Conference on Data Engineering (ICDE), San Diego, California, CA, USA, 2000, pp. 611-621.
10. V. Crescenzi, G. Mecca, and P. Merialdo, "RoadRunner: towards automatic data extraction from large Web sites," Proceedings of the 27th VLDB Conference, Roma, Italy, 2001, pp. 109-118.
11. A. Sahuguet, and A. Fabien, "Building intelligent web applications using lightweight wrappers," Data and Knowledge Engineering, vol. 36(3), pp. 283-316, 2001.
12. Wei Liu, Xiaofeng Meng, Weiyi Meng, "ViDE: A Vision-Based Approach for Deep Web Data Extraction," IEEE, 2010 pp. 447- 460.
13. http://www.w3.org/TR/xpath/ (Accessed: May, 2013).
14. Haikun Hong, Xiaoxin Chen, Guoshi Wu, Jing Li, "Web Data Extraction Based on Tree Structure Analysis and Template Generation," IEEE, 2010, pp. 1-5.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)