INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING & TECHNOLOGY (IJCET)

ISSN 0976 – 6367(Print)
ISSN 0976 – 6375(Online)
Volume 5, Issue 11, November (2014), pp. 57-64

Journal Impact Factor (2014): 8.5328 (Calculated by GISI)
www.jifactor.com

IJCET

UNSUPERVISED APPROACH TO DEDUCE SCHEMA & EXTRACT DATA FROM TEMPLATE WEB PAGES

Mr. Shinde Santaji Krishna[1], Dr. Shashank Dattatraya Joshi[2]

[1]Research Scholar, Department of Computer Engineering, Shri Jagdish Prasad Jhabarmal Tibrewala University, Vidyanagari, Jhunjhunu, Rajasthan, India
[2]Professor, Department of Computer Engineering, Bharati Vidyapeeth Deemed University College of Engineering, Pune, Maharashtra, India

ABSTRACT

Web data extraction has been an important part for many Web data analysis applications. The aim of a web data extraction system is to extract relevant data from the web pages. Embedding of fixed template into web page is done by using fixed template. Thus extracting structured data from the template generated web pages are challenging the task that is useful for Web Information Integration. In this paper, an unsupervised approach is presented that automatically detects schema of web pages & performs page-level extraction task. Here, first visually similar web pages are found out by comparing their visual clues. Then for visually similar web pages, extraction of data is done by using tree merging algorithm that is used in an approach FiVaTech. Also, the schema is detected from the same web pages by removing noisy blocks like advertisement, navigation bars that are irrelevant in a data extraction task. Then time based analysis is performed.

Keywords: Data Extraction, Structured Data, Tree Merging, Web, Wrappe.

1. INTRODUCTION

The World Wide Web is rapidly growing source of information. Structured data on the web are data records generated dynamically from the databases and displayed in web pages using some fixed template. Extracted structured data is used from the template generated pages for information integration and reused in very big range of applications, such as price comparison, product comparison, etc. So, there is a need to automatically extract the structured data from web pages.

Web data extractors are the tools that facilitate extracting relevant data from web pages. Based on automation degree, Web data extraction task can be classified as manually constructed, supervised, semi supervised and unsupervised systems. Manually constructed systems require

programmers to deduct the extraction rules but are costly. Supervised systems require fewer user skills to label the sample pages to induce the extraction rules. To indicate extracted data, semi supervised system do not require users to label any sample pages but require post-processing from the users to choose a pattern. Unsupervised systems generate the wrappers automatically without any user interventions. Based on the extraction, the web data extraction task can be classified as field-level, record-level, page-level and site-level.

This paper presents an unsupervised, page-level data extraction system that automatically extracts data from the web pages. This method is depends on first finding visually and structurally similar web pages to extract the schema and data from the web pages. So, first for given two input web pages we check whether the pages are visually similar or not by making use of VIPS [11] algorithm & fixed/variant template detection algorithm. Then for the visually similar web pages, a schema is detected by using the tree merging algorithm [1] which consists four steps :identification of peer nodes, alignment of matrix, tandem pattern mining, merging of optional nodes .Then data is extracted. Then for the same pages by removing noise blocks, schema is detected by using same tree merging algorithm. Time-based analysis is done based on time required to detect schema by removing noise & without removing noise.

2. RELATED WORK

Reporting of number of approaches is done for extracting information from web pages. Web data extractors deduce extraction rules automatically using supervised techniques or unsupervised techniques. Supervised techniques require the user to provide samples of the extracted data. Unsupervised techniques are that can extract structured data from different web pages without human intervention. WIEN, Soft-Melay represents supervised approaches. There are unsupervised approaches that use DOM tree structure or visual information for extracting the information. Approaches that use DOM tree information are RoadRunner[10], IEPAD[2], MDR[5], DEPTA[7]. Approaches that use visual information are ViPER[14], ViDE[16].

There are two approaches RoadRunner [10] and ExAlg[2] that are closely related which models the template used to generate input documents. RoadRunner is an approach that uses the rule that is initialized to any input document & then it is used to parse another document. ExAlg [2] finds maximal classes of tokens that occur in every input document that corresponds to template and by using token differentiation and nesting criteria it deduces an extraction rule.

3. PROPOSED WORK

Automatically extracting a structured data from the template generated web pages are very tedious task which is used to speed up the data integration. Here a method is proposed to automatically detect a schema & extract data from given web pages. This presents an unsupervised, page-level data extraction approach.

Problem Definition:

Find the schema for the web pages created by some template & extract data.

Schema:

The schema of a website determines structure of a website that identifies the type of data that can be of basic, tuple, set or optional type. For example, for the web page that contains the list of the products by displaying product name, price and discount. Here the product name and price are of basic type whereas discount is of an optional type.

System Overview:

Here, visual clues & DOM tree structure of the web pages are considered to detect the schema of a web page. To detect visually similar web pages, Use of vision-based page segmentation algorithm [11] is done to detect usually similar pages, and visual block tree of each web page is built. Then the visual block trees are compared to check whether the web pages are visually similar or not. For visually similar web pages, their structural similarity can be checked by constructing a fixed/variant pattern tree for each Deep Web site. Application of system is done with either a single or multiple data records in web pages. It detects the schema by making use of identification of peer nodes, alignment of matrix, pattern mining, and merging of optional node [1]. In the peer node recognition step, the same tag in the same level is compared and labeled them with the same symbol if they are matching subtrees. In the matrix alignment step, nodes in the peer matrix are aligned to get a list of aligned nodes. In the pattern mining step, it detects repetitive patterns of different lengths in the aligned list starting from length 1. In the optional node detection step, optional nodes are found out and grouped them if a node disappears in some columns of the matrix. After finding pattern tree, data extraction is done by matching pattern tree & HTML tree at each level. Also for the same web pages, noisy blocks like advertisement, navigation bars are removed & the schema is detected by using tree merging algorithm [1]. Here, analysis is done based on time required to detect the schema & extract data from the web pages without removing noise and by removing noise.

The Architecture

The system architecture is shown in figure 1. Given the input as two web pages, visual similarity of them is checked by applying VIPS [11] and by Match_block algorithm.

For visually similar pages, tree merging algorithm [1] is used to combine DOM trees into a single tree & from leaf nodes of combined tree repetitive nodes are identified. The resultant pattern tree is used to detect the schema of web pages. Also for the same pages, by using noise_block_removal algorithm, creation of DOM trees without noise is done & schema is detected.

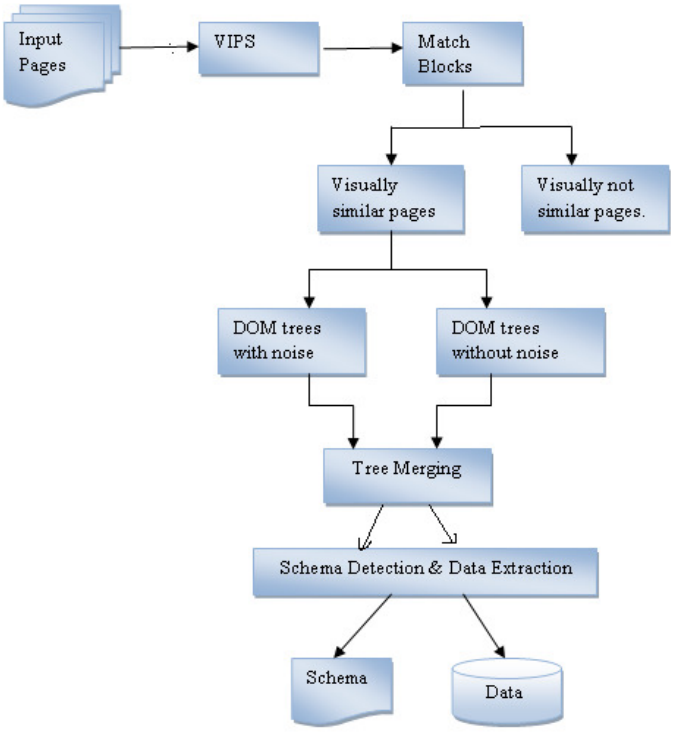

Fig. 1: System Architecture

Algorithm:
Input: Web pages
Output: Schema & extracted data

1 With two input web pages apply VIPS to build visual block trees.
2 Compare visual block trees to check web pages are visually similar or not.
3 If pages are visually similar then
4 If Remove_noise is selected then
5 Apply Remove_Noise algorithm & create DOM trees
6 On DOM, trees apply tree merging algorithm [1] which results in the pattern tree.
7 Identify schema.
8 Compare pattern tree & HTML tree of a web page to extract data.
9 Else
10 On DOM trees without noise, apply tree merging algorithm [1] to create pattern tree.
11 Identify schema
12 Compare pattern tree & HTML tree of a web page to extract data
13 Else
14 Display pages are not visually similar.
15 End.

Steps:

Building visual block trees using VIPS algorithm

We obtain a vision-base content structure of the page by using VIPS (Vision-based Page Segmentation) which combines the DOM structure and visual cues. Using VIPS, a web page is segmented into different blocks to build visual block trees. Extraction of visual blocks, detection of separator, and construction of content structure are three main steps in VIPS. Division of web page is first done into different big blocks, and each big block is segmented until threshold is reached. The threshold is called Permitted Degree of Coherence (PDoC). For each visual block, its DoC is set that detects consistency of contents for each visual block.

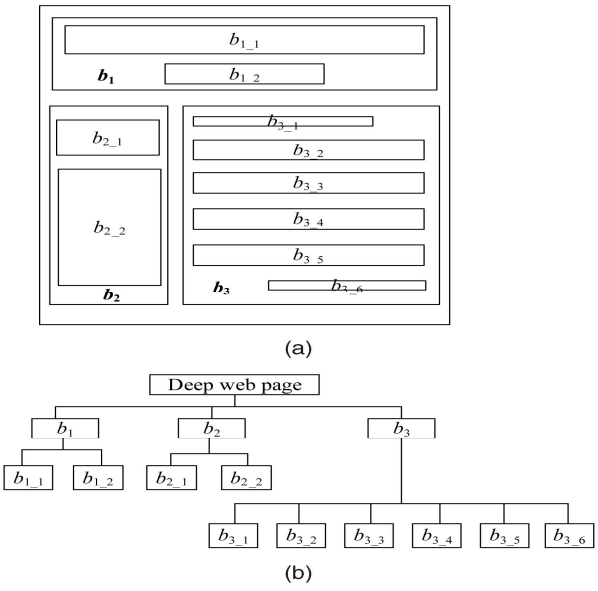

Fig. 2: (a) presentation structure, (b)visual block tree of the page

Detection of visually similar web pages:

To check whether pages are visually similar or not, blocks in visual block trees are compared based on visual features like background color. If the percentage of matched blocks is greater than or equal to the threshold (60%), then pages are visually similar otherwise they are not visually similar.

Algorithm: Match_block (VB1,VB2)
Input: Visual Block trees VB1 & VB2 of web pages
Output: Visually similar or not

1. Initialize threshold=60%;
2. Compare the blocks in VB1 & VB2 based on the visual features
3. If the percentage of matching blocks >= threshold
4. Visually similar pages
5. else
6. Not similar pages
7. EndIf

Noise Block Removal Algorithm

It has become difficult to identify relevant pieces of information since web pages also contain irrelevant content like advertisement, navigation panels. Hence, there is need to remove noise blocks from web pages that help us to get accurate data.

For each tag in DOM tree, there is an associated bounding rectangle. Browser provides position, height, and width of bounding the rectangle of that tag. Calculation of percentage area of each node is done as:

PercentageArea(n)=area(n)/area(<body>)%

Following algorithm is used to remove noise block.[18]

Algorithm Remove_noise (TreeNode)
Input: TreeNode
Output: DOM tree by removing noise blocks
1. Find the area of each child in a TreeNode.
2. if (the area of each child in a TreeNode < 40%) return NULL;
3. biggest = the child with the biggest percentage area;
4. for each child in a TreeNode
5. If (child != biggest) then
6. Remove images from corresponding to tag child
7. Endif
8. Remove_noise(biggest)

Tree Merging Algorithm [1]

For visually similar web pages, the schema is identified by merging DOM trees of the web pages that forms the pattern tree. It consists of the four steps: identification of peer nodes, matrix representation, pattern mining, merging of optional nodes.

Identification of peer nodes:

Here two tags having the same name are compared to check whether they are peer subtrees. Assignment of same symbol is done for peer nodes. Use of two-trees matching algorithm is done to

check similarity. Two subtrees are considered as peer subtrees if the matching score between the two trees is greater than the threshold (0.5).

Matrix Representation :

After the recognition of peer nodes, the next step is to get an associated matrix. The algorithm tries to form an associated matrix such that every row contain the same symbol, or it is basic typed data.

If the row is not aligned in a matrix, then node is shifted depending on the span value of a node. Span value of a node is the highest number of the diverse nodes between any two successive occurrences of the node in each column plus one. Use of following rules is done in order to determine a column to be shifted from the current row r and for identifying required shift distance.

Rules for Matrix Association:

- Rule 1: A column is selected, from left to right, such that a node n with the same symbol exists at upper row r1 & $r\text{-}r_1 < span(n)$.Then that column is shifted with shift distance equal to 1.
- Rule 2: If rule 1 fails, then a column c is selected with nearest row r_{down} from r such that same node is present in a subsequent row or column other than c. In such case, column is shifted with distance $r_{down}\text{-}r$.
- Rule 3: If both rules R1 and R2 fail, and then check if r contains all data (text/image) nodes. In this, no shifting is done.
- Rule 4: If all of R1, R2 and R3 fail, we select the symbol that occurs the maximum number of times on this row. Keeping all columns with that symbol in r unchanged; shift all other columns down by one.

Repetitive Pattern Mining:

This step will detect extra occurrences of discovered pattern & combines them by erasing all occurrences except first one.

Optional Node Merging:

This step identifies optional nodes, the nodes that are missing in some columns of the matrix and group nodes according to their occurrence vector.

Schema Detection [1]

In this step, structure of a website that is a schema is identified. It identifies basic type, set type, optional type & tuple type. It also finds order of set type and optional data.

Data Extraction

After pattern tree is construction, both the HTML trees and the Pattern trees are scanned from top to down simultaneously, matching both at each level and then descending down recursively.

4. EXPERIMENTAL RESULTS

Our system generates the two types of output files, given two web pages of the website as input to the system. The first type presents schema of website in XML-like structure & second type of file (an XML file) presents extracted data.

We performed an experiment on web documents downloaded from the RoadRunner and web sites as given in table.

We performed experiments on a system with Intel Core 2 Duo i3-330 with 3GB RAM. For another measure of the experiment, time-based comparison is conducted between the two systems on following web sites. As shown in the Table, the time consumed by the proposed system is shown in the third column and the time consumed by existing system is shown in the second column. As clear in the table, proposed system is efficient than existing system.

Table 1: Time-based Comparison

Site	Existing System Time with noise (milliseconds)	Proposed System Time without noise (milliseconds)
Kolhapur.nic.in	24843	20079
bluestone	132360	110220
buy	9812	7500
Titan	26984	11890
Maratha shubhvivah	31828	25218
Uefa	50344	44516
pccoepune.com	37719	32015
dblp	7452	6797
ombooks.com	146907	82531
Baresandnoble.com	20031	16297
4accountancyjobs.com	103389	74859

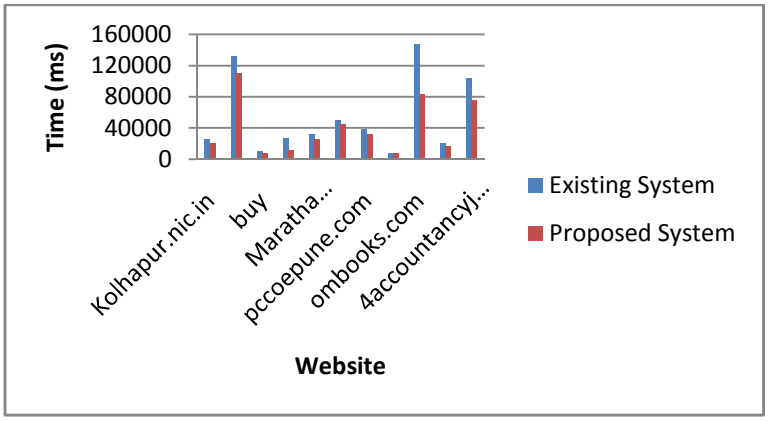

Fig. 2: Time Analysis for Tested Sites

5. CONCLUSION

Here the method is presented to provide an automatic web data extraction. It is capable of extracting the page-level data from visually and structurally similar web pages. This approach uses visual information & DOM tree to extract the schema automatically from the web pages. Here VIPS algorithm & detection of visually similar pages algorithm is used to determine visually similar template pages. Then for visually similar pages pattern tree is constructed by using tree merging algorithm which is used to detect the schema & for extracting data. Schema is detected and data is extracted by removing noise block from the same pages. Time-based analysis is done to show the proposed system is efficient.

REFERENCES

[1] Mohammed Kayed and Chai-Hui Chang, "FiVaTech : Page-Level Web Data Extraction from Template Pages", IEEE transactions on Knowledge and data Engg., vol 22, no. 2, pp. 249-263, Feb. 2010.

[2] A.Arasu and H. Garcia-Molina, "Extracting Structured Data from Web Pages," Proc.ACM SIGMOD, pp. 337-348, 2003.

[3] C.-H.Chang and S.-C. Lui, "IEPAD: Information Extraction Based on Pattern Discovery," Proc.Int'l Conf. World Wide Web (WWW-10), pp.223.

[4] N.Khushmerick, D.Weld, and R.Doorenbos, "Wrapper Induction for Information extraction," Proc. 15th Int'l Joint Conf. Artificial Intelligence (IJCAI), pp. 729-735, 1997.

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

[6] J.Wang and F.H. Lochovsky, "Data Extraction and Label Assignment for Web Databases," Proc.Int'l Conf. World Wide Web (WWW-12), pp. 187-196, 2003.

[7] Y.Zhai and B.Liu, "Web Data Extraction Based on Partial Tree Alignment," Proc.Int'l Conf. World Wide Web (WWW-14), pp. 76-85, 2005.

[8] H. Zhao, W.Meng, Z.Wu, V.Raghavan, and C.Yu, "Fully Automatic Wrapper Generation for Search Engines," Proc.Int'l Conf. World Wide Web (WWW), 2005.

[9] H. Zhao, W.Meng, Z.Wu, V.Raghavan, and C.Yu, "Automatic Extraction of Dynamic Record Sections from Search Engine Result Pages," Proc. Int'l Conf. Very Large Databases (VLDB), pp.989-1000, 2006.

[10] Valter Crescenzi, Giansalvatore Mecca, Paolo Merialdo, "ROADRUNNER: Towards Automatic Data Extraction from Large Web Sites, "Proceedings of the 27th VLDB Conference, Roma, Italy, 2001.

[11] "VIPS: A Vision based Page Segmentation Algorithm", Deng Cai, Shipeng Yu, Ji-Rong Wen and Wei-Yang Ma

[12] J. Wang and F.H. Lochovsky, "Data Extraction and Label Assignment for Web Databases," Proc. Int'l Conf. World Wide Web (WWW-12), pp. 187-196, 2003.

[13] L. Liu, C. Pu, and W. Han, "XWRAP: an XML-enabled wrapper construction system for Web information sources," in Data Engineering, 2000. Proceedings. 16th International Conference on, 2000, pp. 611-621.

[14] K. Simon and G. Lausen, "ViPER: Augmenting Automatic Information Extraction with Visual Perceptions," Proc. Int'l Conf. Information and Knowledge Management (CIKM), 2005.

[15] I. Muslea, S. Minton, and C. Knoblock, "A Hierarchical Approach to Wrapper Induction," Proc. Third Int'l Conf. Autonomous Agents (AA '99), 1999.

[16] W. Liu, X.-F. Meng, W.-Y. Meng. ViDE: A Vision-based Approach for Deep Web Data Extraction. Transactions on Knowledge and Data Engineering, IEEE, 2007

[17] C.-H.Chang, M.Kayed, M.R.Giris ,and K.A.Shaalan, "Survey of Web Information Extraction Systems," IEEE Trans. Knowledge and data Eng., vol 18, no. 10, pp. 1411-1428, Oct. 2006.

[18] Chia-Hui Chang, Chih-Hao Chang, "A Machine Learning Based Approach to Web Extraction from Template Pages", Thesis Advised at Web Intelligence and Data Mining Laboratory.

[19] Rutu Joshi and Priyank Thakkar, "Experimental Evaluation of Different Classification Techniques for Web Page Classification", International Journal of Advanced Research in Engineering & Technology (IJARET), Volume 5, Issue 5, 2014, pp. 91 - 101, ISSN Print: 0976-6480, ISSN Online: 0976-6499.

[20] Alamelu Mangai J, Santhosh Kumar V and Sugumaran V, "Recent Research in Web Page Classification – A Review", International Journal of Computer Engineering & Technology (IJCET), Volume 1, Issue 1, 2010, pp. 112 - 122, ISSN Print: 0976 – 6367, ISSN Online: 0976 – 6375.