ISSN: 2321-7782 (Online)
Volume 2, Issue 3, March 2014
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Paper / Case Study
Available online at: www.ijarcsms.com

Html Tag Based Web Data Extraction and Tree Merging From Template Page

Appukuti Chandrasekhar[1]
Computer Science and Engineering
SVU College of Engineering
Tirupathi – India

Dr. P. Venkata Subba Readdy[2]
Computer Science and Engineering
SVU College of Engineering
Tirupathi – India

Abstract: ***information extraction systems are traditionally implemented as pipeline of special-purpose processing modules targeting the extraction of a particular kind of information.html tag based data are extracting the data usually generated for visualization not for data exchange. Each web page may contain several groups of semi structured data. Each web pages are generated by data values to predefined templates page. Manual data extraction from semi supervised web pages is a difficult task. This paper focuses on study of various automatic web data extraction techniques by using html tag based. There are mainly three types of techniques one is based on automatic extraction and wrapper induction another is page level data extraction. In wrapper induction set of extraction rules are used, which are learnt from multiple pages containing similar data records. Thus extracting information from web pages for searchable websites has been a key step for web information integration.***

Keywords: data extraction, semi structured, html tag, partial tree alignment, wrapper indection.

I. INTRODUCTION

Html tag based web data extraction is time consuming and error prone. In this context automatic web data extraction plays an important role. Example of web data extraction are i) Extract competitor's price list from web page regularly to stay ahead of competition, ii) Extract data from a web page and transfer it to another application iii) Extract people's data from web page and put it in a database.

Automatic extraction is also plays an important role in processing results from search engines. Wrapper is an automated tool that extracts search result records (SRRs) from HTML pages returned by search engines. Database and generated by predefined templates. Extracting such data enables one to collect data from multiple sites and provide services like comparative shopping, meta-querying.

The purpose of this paper is overview of various information extraction techniques like FivaTech[1], EXALG[4], Roadrunner[7], ViPER[8], DeLa[2], DEPTA[3], NET[6], IEPAD[5]. Section II considers more details about above techniques, section III provides a comparison and section IV concludes the paper.

II. WEB DATA EXTRACTION TOOLS

A. DeLa (Data Extraction and Label Assignment for Web Databases):

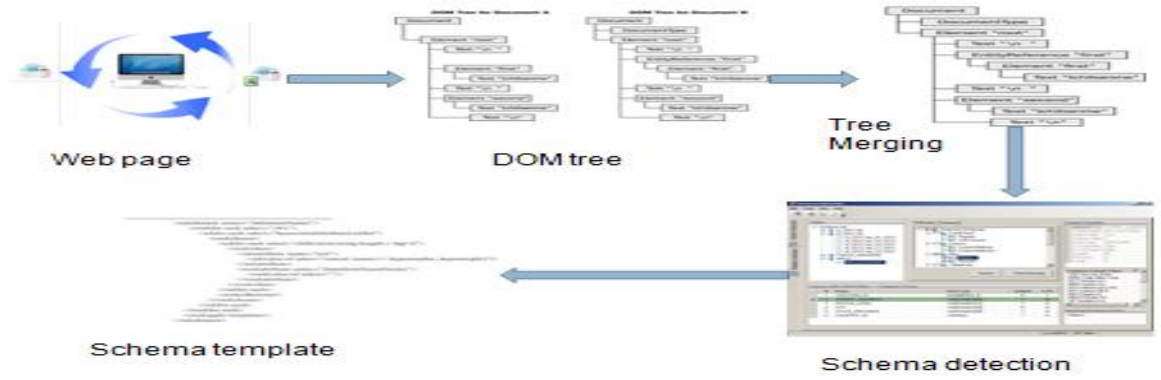
DeLa automatically extract data from web site and assigns meaningful labels to data. This technique concentrates on pages that querying back end database using complex search forms other than using keywords. DeLa system consists of four components: a form crawler, wrapper generator, data aligner, label assigner.

Form crawler: It collect labels of the website form elements. Hidden web crawler HiWe[10] is used for this purpose in DeLa. Wrapper generator automatically generates regular expression wrappers from data contained in pages. Most form elements

contain text that helps users to understand the characteristics and semantics of the element. So form elements are labeled by the descriptive text.

i) Data-rich section extraction

Advertisement, navigational panel are considered as noisy data. These noisy data make data extraction complicated. Noisy data may be wrongly matched with results in inefficient or incorrect wrappers. So it is necessary to identify parts of the page that contain data objects of user interest i.e, data-rich section. Data-rich Section Extraction (DSE) algorithm [11] is used to identify data-rich section. It is performed by comparing two pages of same site. For this traverse DOM trees of the two pages in depth-first order. Each node will be compared and those nodes with identical sub trees at the same depth are discarded.



ii) C-repeated pattern

Structure of data objects appear repeatedly if one page contains more than one data object. These continuous repeated (C-repeated) patterns are discovered as wrapper candidates from token sequences. If a page contains only one data object the data-rich section can be identified by combining multiple pages into single token sequence that will contain multiple data objects.

Definition: - Given an input string S, a C-repeated substring (pattern) of S is a repeated substring of S having at least one pair of its occurrences that are adjacent[2].

Internal structure of a string is exposed by data structure called token suffix-tee [12]. Leaf of suffix tree represented by square with a number. The number indicates the starting position of suffix. Solid circle represents internal node with a number which indicates the token position where its child node differ. Sibling nodes with same parent are arranged in alphabetical order. Label associated with edge between two internal nodes is the sibling between two token positions of the two nodes. Label associated with edge connecting internal node and leaf node is the token at the position of the internal node in the suffix starting from leaf node [2]. Token suffix tree is special suffix tree which can be constructed in O(n) time.

iii) Optional attributes and disjunction

Optional attributes appears once or zero times in a page. Wrapper generator will find out repeated patterns. Among repeated patterns it will select highest nested-level as

Wrapper candidates: There may be multiple patterns with highest nested-level for each page. So number of wrapper candidates may be greater than number of pages in the website. Wrapper candidates may be with some optional missing attributes or some attributes with disjunction values. So there arises a need to construct generalized wrapper from multiple discovered patterns. This can be performed by string alignment. String alignment is performed in O(mn) where n and m are size of two strings S1 and S2

Data Alignment: Data aligner has two phases. They are Data extraction and attribute separation.

i) Data exaction

This phase extracts data from web pages according to the wrapper produced by wrapper generator. Then it will load extracted data into a table. In data extraction phase we have regular expression pattern and token sequence that representing web page. A nondeterministic finite automation is constructed to match the occurrence of token sequences representing web pages.

ii) Attribute separation

Before attribute separation it is needed to remove all HTML tags. If several attributes are encoded in to one text string then they should be separated by special symbol(s) as separator. For instances "@", "$", "." are not valid separator. When several separators are found to be valid for one column, the attributes strings of this column are separated from beginning to end in the order of occurrence portion of each separator.

Label Assignment: To assign labels to the columns of the table containing extracted data four heuristics are employed [2]:

Match from element labels to data attributes. Search for voluntary labels in table header.Search for voluntary labels encoded together with data attributes. Label data attributes in conventional formats.

B. DEPTA (Data Extraction based on Partial Tree Alignment):

DEPTA is a two step approach. First step is to identify data record. Second step extracts data items using partial tree alignment method.

i) Data record extraction

The purpose of this step is to segment the page to identify data records. This step is an enhancement of the MDR technique [14]. MDR algorithm based on two observations about data records in a page and an edit distance string matching algorithm [2]. The two observations are

a). A data record region describes set of similar objects that typically appear in contiguous region of a page and are formatted using same sequence of HTML tags. By considering HTML tags as string we can use string matching

b). This observation is based on the tag tree which is formed by HTML tags in the page. Similar data records should be child sub-trees of same parent node. A web page may contain many data region and different data region have different data records.

Steps in data record extraction

a. Building HTML tag tree: Tag tree is constructed as follows. Find four boundaries of rectangle of each HTML tag by calling embedded parsing and rendering engine of browser [3]. Detect containment relationship between rectangles. Construct tag tree based on containment relationship.

b. Mining data region: This step mines the data region by comparing tag strings of individual nodes including descendants and combination of multiple adjacent nodes. Similar nodes are labeled as data region. Generalized node is introduced to denote each similar individual node and node combination. Adjacent generalized nodes form a data region. Gaps between data records are used to eliminate false node combinations. Visual observations about data records states that gap between the data records in a data region should be no smaller than any gap with in a data record [3].

c. Identifying data

ii) Data extraction: Multiple tag trees of multiple data records are needed to align in order to produce a data base table. In this data table each row represents a data record and column represents data field. This can be performed using multiple alignment method. Partial tree alignment is used in DEPTA. This approach aligns multiple tag trees by progressively growing a seed tag tree. Seed tree is the tree with minimum number of data fields that is picked initially. The selection of seed tree should be in such a way that it should have a good alignment with data fields in other data records. For each tree Ti[$i \neq s$] the algorithm tries

to find a matching node in Ts. When a match is found for node ni, a link is created from ni to ns to indicate its match in the seed tree. If no match found then algorithm attempts to expand the seed tree by inserting ni in to Ts. The expanded seed tree is used for subsequent match.

C. ViPER (Visual perception based Extraction of Records):

It is a fully automated information extraction tool. This technique is based on assumption that the web page contains at least two consecutive data records which exhibits some kind of structural and visible similarity. ViPER is able to extract relevant data with respect to user's visual perception of the web page. Multiple Sequence Alignment (MSA) technique is used to align these relevant data regions.

Data Extraction:

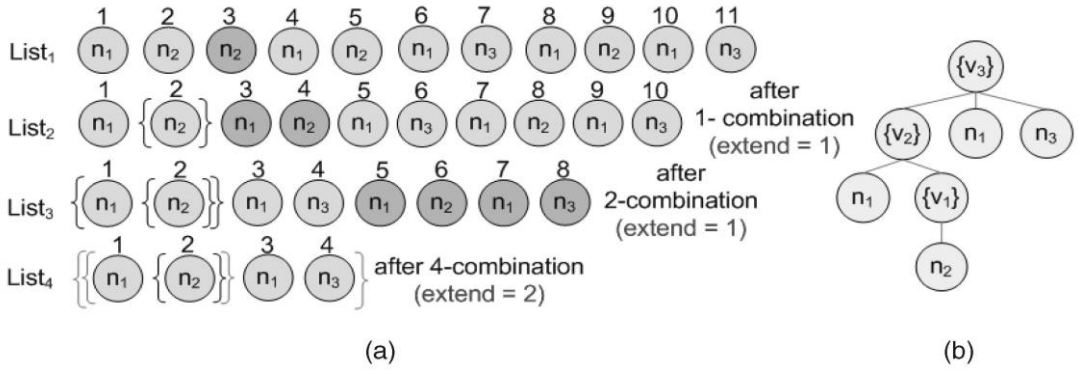
Preprocessing is performed to improve pattern extraction accuracy. Preprocessing provides the ability to access parsed document tree T* with additional rendering information. Every tag element is augmented with bounding box information by the upper left corner's (x,y) pixel coordinates along with width and height. For analysis abstract representation of T* is created in which each HTML tag is restricted to tag name ignoring attributes. Text between two tags represented by a new element denoted as <TEXT> element tag. Preprocessed document is called restricted tag tree T and plain tag sequence structure S of T where each element in the tree has a link to the corresponding element in the sequence representation and vice versa [8].

Pattern search: Similarity between two plain sequences Si, Sj with length m, n respectively is measured using technique edit distance. One disadvantage with edit distance is that repetitive and optional subparts inside the sequence Si, Sj should increase edit cost, so possible matches may be discarded. These optional subparts are handled by similarity threshold value θ. Two sequences will be similar if their accumulated edit distance is less or equal to threshold value.

Primitive tandem repeats: Tandem repeat contained in a sequence is a subpart of S. Tandem repeat construct an array of consecutive repeats. A repeat is primitive if it does not contain shorter repeats. Each extra repetitive instance will be marked with different marker elements. According to these marked tag elements the recursive computation of a single matrix entry of D is adapted.

ii). Data Alignment:

Sub-optimal solution can be obtained using center- star tree alignment algorithm which finds a sequence (center sequence) minimizing overall edit cost to each remaining sequences. OLERA [19] is a semi supervised information extraction tool where user can generate extraction rules according to training pages. There are several drawbacks in center star technique [8]. This can be overcome by using global sequence alignment which uses general suffix tree.



D. ROAD RUNNER:

Road runner defines data extraction problem as "given a set of sample HTML pages belonging to the same class, find the nested type of the source data set and extract the source data set from which the pages have been generated"[7].

Matching Technique: Matching technique is based on ACME (Align, collapse under mismatch and Extract) technique. There are two assumptions i) tags are properly closed ii) source pages are transformed into tokens by a lexical analyzer. Matching algorithm works on list of tokens and wrappers. Wrapper for each page will be created initially. Then they are progressively refined to find common regular expression among them. This can be performed by solving mismatches between wrappers and tokens. Mismatch occurs when token does not comply with grammar specified by the wrapper at the time of parsing. Usually there are two types of mismatches. String mismatches and tag mismatches. String mismatches happen when different strings occur at same position of wrapper. This is solved by generalizing wrapper by replacing newly discovered fields by same symbol. String mismatches are used to discovering fields. This is performed in three steps.

i) Square location by Terminal-Tag search :

One hint about the square is that both wrapper and sample contain at least one occurrence of square. If Ow and Os are no of occurrences of square in wrapper and token respectively, then we can assume that min(Ow, Os) occurrences have been matched. Last token of square i.e, the token immediately before mismatch, is called terminal tag. There are two possibilities. Candidate square may be in wrapper or in sample. This can be checked by searching both sample and wrapper.

ii) Square Matching

This verifies whether candidate tokens really form a square. This is done by searching the candidate square backwards. If search succeeds then we can conclude it is a square.

iii) Wrapper generation

If S is an optional tag then it can be represented as (S)? . If S is a square then it can be represented as (S)+.

E. EXALG:

EXALG performs template extraction in two stages. First stage is Equivalence Class Generation Stage (ECGM) and second is analysis stage as shown in Fig. 2. ECGM stage computes equivalence classes. i.e, set of tokens having same frequency of occurrence in every page. This is performed by FindEquiv sub module. There may be many equivalence classes. But EXALG only considers equivalence classes that are large and contain tokens which occur in large number of pages. Such equivalence classes are called Large and Frequently Occurring Equivalence Classes(LFEQs). It is very unlikely for LFEQs to be formed by chance. LFEQs are formed by tokens associated with the same type constructor in the template[4].

F. NET (Nested data extraction using Tree matching and Visual cues) :

NET extract data items from data records even it handles nested data records also. There are two main steps. Building tag tree is difficult because page may contain erroneous and unbalanced tags. This is performed based on nested rectangles. For this four boundaries of each HTML element are determined by calling embedded parsing and rendering engine of a browser. A tree is constructed based on containment check, whether one rectangle contained inside another.

G. FivaTech :

FivaTech is a page-level web data extraction technique. Data extraction is performed in two modules. First module takes DOM trees of web pages as input and merges all DOM trees into a structure called fixed/variant pattern tree. In the second module template and schema are detected from fixed/variant pattern tree. Fig. 3 shows FivaTech approach.

First module arranges all nodes of input DOM trees into a matrix form. This module can be divided into four sub modules. They are Peer node recognition, multiple string alignment, Pattern mining, Optional node merging. Nodes which have same tag name but different functions are called peer nodes. Peer nodes are denoted using same symbol in order to facilitate string alignment. Pattern mining on aligned string will remove extra occurrences of discovered pattern.

Peer node recognition: Peer nodes are identified and they are assigned same symbol. Simple Tree Matching [STM] algorithm together with score normalization [1] is used for identifying peer nodes.

Matrix alignment: This step aligns peer matrix to produce a list of aligned nodes. Matrix alignment recognizes leaf nodes which represent data item.

Optional node merging: This step recognizes optional nodes, the nodes which are which disappears in some column of the matrix. This step groups nodes according to their occurrence vector.

Schema detection module detects structure of the website i.e, identifying the schema and defining the template. The items contained in a page can be divided into basic type, set type, optional type and tuple type [1]. This step recognizes tuple type as well as order of set type and optional data which are already identified by previous module.

III. COMPARISON

Among the webpage extraction techniques discussed above, some techniques reveals flat records and some other techniques are trying to extracts nested records also. DEPTA and NET will find out nested records in addition to flat records. All other techniques produce only flat records. DeLa, RoadRunner and IEPAD extracts records using wrapper induction method, others are based on operations on tree structure of the page such as tree alignment, tree merging and tree matching. In DEPTA extraction is performed mainly by partial tree alignment. FivaTech uses tree merging technique whereas NET using tree matching. Other extraction methods are based on visual perception, equivalence class generation which is used by ViPER and EXALG respectively. RoadRunner, EXALG and FivaTech considers multiple pages of website and other techniques considers only single page. Summary of comparison is shown in Table I.

IV. CONCLUSION

Html tag based, this paper studied various approaches to extract structured data from web pages. Some of these techniques are either inaccurate or make many strong assumptions. These techniques reconstructs hidden back-end database. Some techniques use regular expression wrappers to extract data.

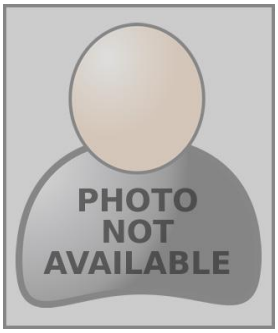
Acknowledgement

I wish to express my hearty gratitude and sincere regards to authors Mohammed Kayed and Chaia-Hui Chang, Senior Member, IEEE, and Chee Keong Chan and Jiawei Han and Micheline Kamber having provided me valuable information about the progress in my work, interms of their paper publications and text book.

References

1. J. Wang and F.H. Lochovsky, “Data Extraction and Label Assignment for Web Databases,” Proc. international conference on World Wide Web (WWW-12), pp. 187-196, 2003.
2. Y. Zhai and B. Liu, “Web Data Extraction Based on Partial Tree Alignment,” Proc. international conference on World Wide Web (WWW-14), pp. 76-85, 2005.
3. A. Arasu and H. Garcia-Molina, “Extracting Structured Data from Web Pages,” Proc. ACM SIGMOD, pp. 337-348, 2003.
4. C. H. Chang and S. C. Lui, “IEPAD: Information Extraction Based on Pattern Discovery,” Proc. International Conference on World Wide Web (WWW-10), pp. 223-231, 2001.
5. Bing Liu and Yanhong Zhai, “NET - A System for Extracting Web Data from Flat and Nested Data Records,” Proc. WISE'05 Proceedings of the 6th international conference on Web Information Systems Engineering, pp. 487-495, 2005.

AUTHOR(S) PROFILE



Appukuti Chandra Sekhar received B.Tech degree in Computer Science and Information Technology from Priyadarshini Engineering College, JNTUA University, Anantapuram, A.P, India in 2011 and currently pursuing M.Tech, Computer Science and Engineering, final semester, from Sri Venkateswara University College of Engineering, TIRUPATI, A.P, India. Her interested areas are Data Mining, Software Engineering, and Software Architecture.

Dr. P. Venkata Subba Reddy working as a Professor of Computer Science and Engineering, Sri Venkateswara University, Tirupathi, A.P, India. He completed MSc and MPhil. He completed his Ph.D in the area of Aritificial Intelligence. He has more than 28 years of teaching experience. He published many papers in the peer-refereed journals and conferences. His interested areas are Aritificial Intelligence, Cloud computing, and Data Mining.

