International Journal of Research In Science & Engineering e-ISSN: 2394-8299
Volume: 3 Issue: 4 July-August 2017 p-ISSN: 2394-8280

A SURVEY ON WEB DATA EXTRACTION TECHNIQUE

Nitika Kadam[1]
Asst. Prof.,Malwa Insitute of Science & Technology,Indore
Kadamnitika01@gmail.com

ABSTRACT

Web pages are usually generated for visualization not for data exchange. Each page may contain several groups of Structure data. Web pages are generated by plugging data values to predefined templates. Manual data extraction from semi supervised web pages is a difficult task. This paper focuses on study of various automatic web data extraction techniques. There are mainly two types of techniques one is based on wrapper induction another is automatic extraction. In wrapper induction set of extraction rules are used, which is learnt from multiple pages containing similar data records.

Keywords - Data extraction, wrapper induction, DOM tree, web crawler, Data alignment, pattern mining.

INTRODUCTION

Internet is a powerful source of information. Most business applications depend on web to collect information that is crucial for decision making process. By analyzing web we can identify market trends, price details, and product Specification etc. Manual 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. Websites are usually designed for visualization not for data exchange. All pages of same website will be well designed. They may follow same template. Templates can be used to display objects of same type. Web page construction is the process of combining data to templates. Web data extraction is the reverse process of page generation. If multiple pages are given as input the extraction target will be the page wide information. If one page is given as input extraction target will be record level information. 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. Automated extraction is easier with Google and Amazon because theyhave web service interfaces. But search engines that support business to customer applications does not have web service interfaces. Search engine result contains query independent (static contents), query dependent (dynamic) contents, contents affected by many queries but independent of content of specific query(semi-dynamic). Typically dynamic sites are filled with data from back-end 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 DeLa ,FivaTech[1], EXALG[4].

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. Fig. 1 shows the architecture of DeLa.

***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. These labels are used further to compare with attributes of data extracted from query-result page.

***Wrapper Generation*:** Pages gathered by the form crawler are given as input of the wrapper generator. Wrapper generator produce regular expression wrapper based on HTML tag structures of the page. If a page contains more than one instance of data objects then tags enclosing data objects may appear repeatedly. Wrapper generator considers each page as a sequence of tokens composed of HTML tags. Special token "text" is used to represent text string enclosed with in HTML tag pairs. Wrapper generator then extracts repeated HTML tag substring and introduces a regular expression wrapper according to some hierarchical relationship between them. Techniques used in wrapper generator are

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 n 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 differs. 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. In order to discover C-repeated patterns path-labels of all internal nodes and their prefixes in the token suffix-tree are considered as candidates. A candidate repeated pattern is a C-repeated pattern if any two of its occurrences are adjacent. Repeated patterns are adjacent if the distance between two starting positions is equal to the pattern length. Algorithm that discovers all C-repeated patterns from a suffix tree in O (n log n) time is presented in [13]. For discovering nested structures a hierarchical pattern tree is used. A pattern tree can be used to represent dependence and independence between discovered C-patterns.

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 (m n) where n and m are size of two strings S1 and S2.

B 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.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, and 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.

C. IEPAD

IEPAD is an information extraction system which applying pattern discovery techniques. It has three components, an extraction rule generator, pattern viewer and an extractor module. Extraction rule generator accepts input web page and generate extraction rules. Extraction rule generator includes a token translator, PAT tree constructor, pattern discoverer, a pattern validator and an extraction rule composer as shown in Fig. 4. Pattern viewer is a graphical user interface which shows the repetitive pattern discovered. Extractor module extracts desired information from pages. Extraction rules generated by extraction rule generator can be used by the extractor module to extract information from other pages which are following similar structure. Translator generates tokens from input webpage. Each token is represented by a binary code of fixed length l. PAT tree constructor receives the binary file to construct a PAT tree. PAT tree is a PATRICIA tree (Practical Algorithm to Retrieve Information Coded in Alphanumeric [21]). PAT tree is used for pattern discovery. Discoverer uses PAT tree to discover repetitive patterns called maximal repeats. Validator filters out undesired patterns from maximal repeats

and generates candidate patterns. Rule composer revises each candidate pattern and to form an extraction rule in regular expression [5].

II. 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 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.. 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.

IV CONCLUSION

This paper studied various approaches to extract Structure 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 objects.

V REFERENCES

[1] Mohammed Kayed and Chia-Hui Chang. "FiVaTech: Page- Level Web Data Extraction from Template Pages," IEEE Transactions on Knowledge and Data Engineering, VOL. 22,
NO. 2, 2010.
[2] 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.
[3] 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.
[4] A. Arasu and H. Garcia-Molina, "Extracting Structured Data from Web Pages," Proc. ACM SIGMOD, pp. 337-348, 2003.
[5] 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.
[6] 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.
[7] V. Crescenzi, G. Mecca, and P. Merialdo. ROADRUNNER: Towards automatic data extraction from large web sites. In Proc.
of the 2001 international conference on Very Large Data Bases,pp 109–118, 2001.
[8] K. Simon and G. Lausen, "ViPER: Augmenting Automatic
Information Extraction with Visual Perceptions," Proc.
International Conference on Information and Knowledge
Management (CIKM), 2005.
[9] H. Zhao, W. Meng, Z. Wu, V. Raghavan, and C. Yu,

e-ISSN: 2394-8299
p-ISSN: 2394-8280

"Automatic Extraction of Dynamic Record Sections from Search Engine Result Pages," Proc. international conference on Very Large Databases (VLDB), pp. 989-1000, 2006.
[10] S. Raghavan and H. Garcia-Molina. "Crawling the hidden web," Proc. 27th VLDB Conf., 2001, 129-138.