Web Scraping Framework based on Combining Tag and Value Similarity

Shridevi Swami[1], Pujashree Vidap[2]

[1] Department of Computer Engineering, Pune Institute of Computer Technology, University of Pune, Maharashtra, India
[2] Department of Computer Engineering, Pune Institute of Computer Technology, University of Pune, Maharashtra, India

Abstract

When user fires a query on data intensive web sites, the response to the user query is web page generated dynamically, consisting of Query Relevant Records (QRRs). Along with user desired data these QRRs are decorated with irrelevant data such as advertisements, navigational panels etc. Deciding which region of this result page contains the relevant data is easy for human but not for computer programs. Thus, for utilization of this data removal of irrelevant data and atomic extraction of QRRs from result web pages is necessary , which further can be used in value added services like comparison shopping, data integration, meta querying etc.

This paper discusses various atomic data extraction techniques and proposes a new approach which uses similarity of Tag and Value together to extract QRRs automatically from query result page and aligns the extracted QRRs in structured format e.g. tables where they can be easily aggregated and compared. The challenge of proposed automatic data extraction is to handle the situation when QRRs are not contiguous as query result page often contains auxiliary query irrelevant information and that of data values alignment present in the extracted records into a table so that the data values for the same attribute in each record are placed into the same column in the table.

Keywords: *Data aggregation, Data integration, Data scraping, Data values alignment, Wrapper.*

1. Introduction

In today's world the most powerful source of information is World Wide Web. People can use tools such as search engines to get the desired information from the web. There is requirement of interaction with search engines from both web users and web applications.

Many business applications have to depend on web for decision making because information is to be aggregated from different web sites. One can analyze and summarize the collected web data and can use it to find recent trends in market, details of price, product specification details etc. Manually performed data extraction consumes large time and leads to more errors. In this perspective an important role is played by automatic web data extraction. So we build wrapper, an automated tool which extracts Query Relevant Records (QRRs) from HTML pages returned by search engines. Normally Search engine result consists of query independent contents i.e. static contents, query dependent contents i.e. dynamic contents, while some contents are semi-dynamic.

The main motivation behind automatic data extraction is that it is a key element of applications, like meta-querying, data integration and comparison shopping, where multiple web databases are queried to collect data from multiple sites and provide value added services. Further, if we aggregate the extracted data values in structured format like tables then it is very easy to compare them. So, this fact leads to the alignment of data values present in the extracted QRRs into a table.

In this paper, we propose Web Scraping framework which uses Tag and Value similarity together for automatically extracting data from query result pages by first identifying and then segmenting the Query Relevant Records (QRRs) and then aligning the segmented QRRs into a table such that the data values of the same attribute in each QRR are put under the same column of the table.

The formation of the paper is given as Section II discusses related work, Section III discusses proposed Web Scraping framework approach, Section IV discusses the mathematical model of proposed approach and Section V concludes the paper.

2. Related Work

Web data extraction system, automatically and repeatedly extracts data from dynamic web pages and can deliver the extracted data to a database or some other application.

Web data extraction system is a wrapper generator which mainly designed with basically three approaches [2]: First, Manual wrapper programming languages, second is Wrapper induction and third is Automatic wrapper generation.

2.1 Manual wrapper programming languages

This approach supports special pattern specification languages to help the user to construct extraction programs. To hide their complexities visual platforms are provided under simple graphical wizards and interactive processes [9].

2.2 Wrapper Induction Method

These methods require human assistance to build a wrapper. In wrapper induction, extraction rules are derived based on inductive learning. The items present in a set of training pages are labeled or marked by the user to extract the intended items, and then system uses the labeled data to learn the wrapper rules and further uses them to extract records from new pages. A rule is made up of two patterns called as prefix and suffix pattern, which represents, respectively beginning and end of the target item.

2.3 Automatic Extraction Method

To overcome the problems of wrapper induction, some unsupervised learning methods, have been proposed to automatically extract the data from the query result pages. These methods fully depend on the tag structure present in the query result pages.

2.3.1 DeLa [5]

Features:

1) It models the structured data contained in template-generated web pages as string instances encoded in HTML tags, of the implied nested type of their web database.
2) A regular expression is employed to model the nested HTML-encoded version. Since the HTML tag-structure which encloses the data may appear repeatedly if the page contains more than one instance of the data, first the page is changed into a token sequence made up of HTML tags and a special token ”text” representing any text string enclosed by pairs of HTML tags. Then, continuously repeated substrings are extracted from the token sequence and a regular expression wrapper is induced from the repeated substrings according to some hierarchical relationships among them.

2.3.2 ViPER [7]

Features:

1) ViPER takes the advantage of both visual data value similarity features and the HTML tag structure to identify and rank possible repetitive patterns.
2) Then, alignment of matching subsequences is done with global matching information.

2.3.3 ViNTs [4]

Features:

1) Using both visual and tag features ViNTs learns a wrapper from a set of training pages from a website.
2) It first uses the visual data value similarity without considering the tag structure to identify data value similarity regularities, denoted as data value similarity lines, and then combines them with the HTML tag structure regularities to generate wrappers.
3) Both visual and non visual features are used to weight the relevance of different extraction rules.
4) The resulting wrapper is represented by a regular expression of alternative horizontal separator tags (i.e., <HR> or <BR> <BR>), which segment descendants into QRRs.

2.3.4 DEPTA [2]

Features:

1) It identifies data records without extracting data items in the records by using method based on visual cues.
2) Further by using partial tree alignment technique it aligns corresponding data items from multiple data records and puts the data items in a database.

The Table1, Table2 and Table3 showed below gives the comparison of these three approaches with respect to their examples, advantages and disadvantages.

Table 1: Examples

Wrapper Generation Method	*Examples*
Manual wrapper programming languages	WICCAP, Wargo, Lixto, etc.
Wrapper induction	WIEN, Soft Mealy, WL Stalker, XWRAP, etc.
Automatic wrapper generation	DeLa, ViPER, ViNTs, DEPTA, etc.

Table 2: Advantages

Wrapper Generation Method	*Advantages*
Manual wrapper programming languages	1) Very expressive extraction programs can be created.
Wrapper induction	1) It uses supervised learning to learn data extraction rules from a set of manually labeled examples. 2) As the user labels only the items of interest, extra data not at all extracted.
Automatic wrapper generation	1) It automatically identifies data record boundaries.

Table 3: Disadvantages

Wrapper Generation Method	*Disadvantages*

Manual wrapper programming languages	1) Difficult to use for professionals.
Wrapper induction	1) It requires labor intensive and time-consuming manual labeling of data. 2) It is not scalable to a large number of web databases. 3) When the format of a query result page changes existing wrapper gives poor performance. 4) There is need of monitoring changes in a page's format and maintaining a wrapper when a page's format changes.
Automatic wrapper generation	1) DeLa often generates multiple patterns (rules) and it is hard to decide which one is correct. 2) ViPER suffers from poor results for nested structured data. 3) Multiple result pages with at least four QRRs, and one no result page is basic requirement of ViNTs wrapper. 4) As ViNTs uses prelearned wrapper, continuous monitoring of changes in format of query result pages is required, this is difficult one. 5) DEPTA derives wrappers fully based on HTML tags which may not always produce the accurate wrapper.

3. Proposed Approach

The objective of proposed Web Scraping framework is to automatically extract the Query Relevant Records (QRRs) in a page, and align the data values of the QRRs into a table.

As shown in Fig. 1, the input to the system is query result page containing atleast two QRRs which passes through two steps i.e. data extraction and data alignment to give output as QRRs and aligned data values respectively.

When a query result page is given, the Tag Tree Construction module first constructs a tag tree for the page rooted in the <HTML> tag. Then, in the Data Region Identification module, identification of all possible data regions is done, which usually contain dynamically generated data, top down starting from the root node. When system enters into the Record Segmentation module it segments the identified data regions into data records according to the tag patterns in the data regions. Finally, when the segmented data records are given, the Query Result Section Identification module is responsible for selecting one of the data regions as the one that contains the QRRs. Further given the QRRs Pairwise Alignment module aligns the data values of QRRs which further holistically aligned by Holistic Alignment module.

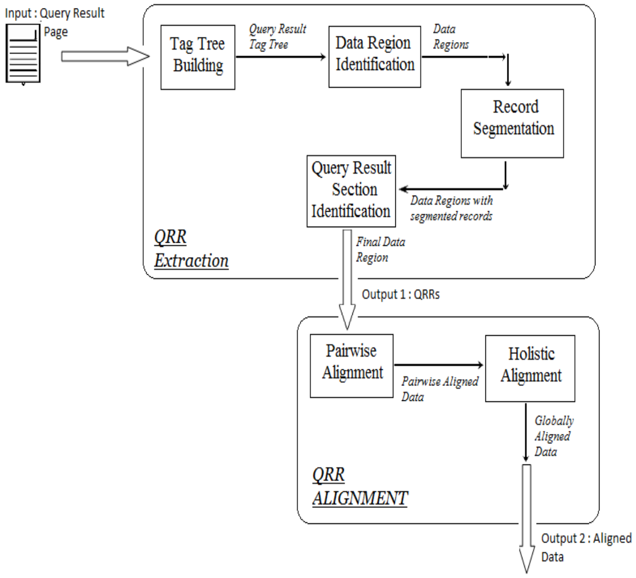

Fig. 1 Proposed architecture for Web Scraping Framework

4. Mathematical Model

The proposed approach can be modeled mathematically using set theory as follows.
Let S be a system.
S = {I, F, O}
Where 'I' is set of Inputs, 'F' is set of functions, 'O' is set of Outputs.

Input:
$I = \{Q_i\}$
Where Q_i is set of query result Web Pages WP_i containing query relevant as well as irrelevant data.
i.e. $Q_i = \{WP_1, WP2, \dots, WP_i\}$

Functions:
$F = \{F_1, F_2, F_3, F_4, F_5, F_6, F_7\}$
Where F_1 = Preprocessing
F_2 = Tag tree construction
F_3 = Data region identification
F_4 = Record segmentation
F_5 = Query result section identification
F_6 = Pairwise alignment
F_7 = Holistic Alignment

Output:
$O = \{O_1, O_2, O_3\}$
Where O_1 = Extracted QRRs
O_2 = Pairwise aligned data values of each pair of QRR
O_3 = Globally aligned data values of all QRRs
Now, let us elaborate each function in detail

1. Let $F_1 = \{I_{11}, F_{11}, O_{11}\}$

Where $I_{11} = \{WP \mid WP \in Q_i\}$
F_{11} is defined as,
T = T1 - T2, where T is finite set of Required Tags in WP, T1 is finite set of All Tags in WP and T2 is set of Removable Tags such that
T2={<script>,<input>,….<meta>}
$O_{11} = \{P_i\}$ where P_i is preprocessed query result pages containing query relevant as well as irrelevant data.

2. Let $F_2 = \{I_{12}, O_{12}\}$

Where $I_{12} = O_{11}$,
$O_{12} = \{V, E\}$
Where $V=\{R \square V_T \square V_c\}$
Where R is root of tag tree,
$\forall$ V_T is finite set of tag nodes (internal nodes)
i.e. $\forall$ $V_T = \{V_{T1}, V_{T2}... V_{Tn}\}$
Where $\forall$ $V_{Tn} = \{t_n, ts_n, sib\_id, reg\_id\}$,
V_c is finite set of content nodes (leaf nodes)

3. Let $F_3 = \{I_{13}, F_{13}, O_{13}\}$

Where $I_{13} = O_{12,}$
F_{13} is similarity function defined as similarity (n_i, n_j) where n_i and n_j are level wise paired nodes in top down manner of tag tree O_{12}
i.e. similarity (ts_{ni}, ts_{nj}) where ts_{ni} and ts_{nj} are tag strings of n_i and n_j and similarity(n_i, n_j) $>=T_{nsim}=0.6$
O_{13} can be defined as set of Regions $R= \{R_1, R_2,…. R_i\}$
where, $\forall$ $R_i = \{R_{id} , R_{nodes} ,R_{node\_sid} \}$
and $R_{nodes} =\{n_1,n_2,…n_n|$ similarity$(n_i, {}_{nj})$ $>=0.6\}$

4. Let $F_4 = \{I_{14}, F_{14}, O_{14}\}$

Where $I_{14} = O_{13}$
F_{14} is function on each $R_i =O_{13}$ defined as,

$$R_{nodes} \text{ X } R_{pattern} = R_{TandemRepeat}$$

O_{14} can be defined as set of Segmented Regions SR
i.e. SR= $\{SR_i \mid i=1,2,…n\}$
Where $\forall$ $SR_i = \{R_{id}, R_{pattern}, R_{TandemRepeat}\}$

5. Let $F_5 = \{I_{15}, F_{15}, O_{15}\}$

Where $I_{15} = O_{14}$
F_{15} finds the TotalWeight (SR_i),
$SR_i = \sum$ (AreaWeight$_i$ + CenterDistWeight$_i$ + DataStringsWeight$_i$)
$O_{15} = \{SR_k\}$ where SR_k is final data region such that

$$\text{TotalWeight }(SR_k) = \max_{i=1,2,\ldots n} \sum \text{TotalWeight }(SR_i) \text{ where } i=1,..n$$

6. Let $F_6 = \{I_{16}, F_{16}, O_{16}\}$

Where $I_{16} = \{O_{15} , O_{14}\}$
F_{16} creates,
1) $QRR_1 = \{f1_1,….f1_i\}$,………$QRR_n=\{ fn_1,….fn_j \}$ where fi_j refers to j^{th} data value of r_i.

2) For each pair of QRR calculate data value similarity 's' of each pair of data value f_i,f_j as follows where DT is data type.
If (DT $(f_{i,\ fj})$ = Interger) then s=1;
If (DT (f_i, f_j) = Double) then s=1;
If (DT (f_i, f_j) = Price) then s=1;
If ((DT (f_i) = Integer) AND (DT (f_j) =Double) OR (DT (f_i) =Double) AND (DT (f_j) =Integer)) then s=0.5;
If (DT (f_i,f_j) = String) then s=cosine similarity (f_i,f_j)
If (DT (f_i) != DT(f_j)) then s=0;
3) The data values are alignment with following constraints
a) Each data value can be aligned to at most one data value from the other QRR.
b) Cross alignment cannot be performed.
O_{16} can be defined as set of Tables
$O_{16}= \{T_1, T_2, .. T_i\}$ where $\forall$ T = {Data values, column number}

7. Let $F_7 = \{I_{17}, F_{17}, O_{17}\}$

Where $I_{17} = O_{16}$, Pairwise alignment represented as an undirected graph G=(V,E) where
V= {Each data value 'f' in QRRs}
E= {Present if pairwise alignment exist between two data values}
F_{17} finds nonintersecting Connected Component of G as $\{CC_1,CC_2,…..,CC_n\}$ where $\forall$ $CC_n =\{d_1,d_2,…,d_i\}$ where $\forall$ d_i is data values such that $d_i \in V$ and Each connected component of the graph represented as table column inside which the connected data values from different records are aligned vertically.
O_{17} can be defined as Holistically aligned table O_{17}= T where T is set of Columns
i.e. T= $\{C_1,C_2,…..,C_n\}$ where $\forall$ $C_n = CC_n$

5. Conclusions

This paper discusses the framework for Web Scraping. This approach uses both Tag and Value similarity unlike previous methods, for QRR data extraction which can be contiguous or non contiguous. Also this service can be used in data integration application where data from two different web pages is extracted and presented in structured format e.g. in table by using new alignment techniques called Pairwise and Holistic alignment, which can be further used for value added services like comparison shopping .

References

[1] B. Liu, R. Grossman, and Y. Zhai, "*Mining Data Records in Web Pages*", Proc. Ninth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pp. 601-606, 2003.

[2] B. Liu and Y. Zhai, "*Structured Data Extraction from the Web Based on Partial Tree Alignment*" , IEEE Trans.

Knowledge and Data Eng., vol. 18, no. 12, pp. 1614-1628, Dec. 2006.

[3] F. H. Lochovsky, W. Su, J. Wang and Yi Liu, "*Combining Tag and Value Similarity for Data Extraction and Alignment"* , IEEE Trans. Knowledge and Data Eng., vol. 24, no. 7, pp. 1186-1200, July. 2012.

[4] H. Zhao, W. Meng, Z. Wu, V. Raghavan, and C. Yu, "Fully Automatic Wrapper Generation for Search Engines," Proc. 14th World Wide Web Conf., pp. 66-75, 2005.

[5] J. Wang and F.H. Lochovsky, "Data Extraction and Label Assignment for Web Databases," Proc. 12th World Wide Web Conf.,pp. 187-196, 2003.

[6] J. Wang and F. Lochovsky, "*Data-Rich Section Extraction from HTML Pages"*, Proc. Third IntSl Conf. Web Information System Eng., 2002.

[7] K. Simon and G. Lausen, "*ViPER: Augmenting Automatic Information Extraction with Visual Perceptions"*, Proc. 14th ACM Int'l Conf. Information and Knowledge Management, pp. 381-388, 2005.

[8] M. Alvarez, A. Pan, J. Raposo, F. Bellas, and F. Cacheda, "Using Clustering and Edit Distance Techniques for Automatic Web Data Extraction", Springer-Verlag Berlin Heidelberg, pp. 212–224, 2007.

[9] R. Baumgartner, S. Flesca, and G. Gottlob, "*Visual Web Information Extraction with Lixto"* , Proc. 27th Int'l Conf. Very Large Data Bases, pp. 119-128, 2001.

Shridevi Swami graduated from Pune University. She is lecturer in department of Information Technology at VPCOE Baramati, Pune. She is also Postgraduate student at PICT Pune.

Pujashree Vidap obtained postgraduate degree from Pune University. She is Assistant Professor in Computer department at PICT Pune.