Volume 3, Issue 4, April 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Towards Automatic Data Extraction Using Tag and Value Similarity Based on Structural -Semantic Entropy

P.V.Praveen Sundar[1]

[1]*Research Scholar,*
Hindusthan College of Arts &Science,
Coimbatore,India.

Abstract: An automatic web record extraction extracts a set of objects from heterogeneous web pages based on similarity measure among objects in an automated fashion. This classifies a region in the web page according to similar data object which emerge frequently in it. This involves transformation of unstructured data into structured data that can be stored and analyzed in a central local database. The existing system develops a data extraction and alignment method known as combining tag and value similarity (CTVS), which identifies the query result records (QRRs) by extracting the data from query result page and segment them. Those segmented QRRs are aligned into a table where same attribute data values are put into the same column. This technique is based on the discovery of non consecutive data records to detect nested data records in QRRs. Those attributes in record are aligned using record alignment algorithm by combining the tag and data value similarity information based on similarity measure. Besides the structure of the data value is altered when extracting from the webpage. Those changes in template make it inefficient to properly access them as done in traditional databases. The proposed structural semantic entropy measures the degree of repeated occurrence of information from DOM tree representation. This aims to locate the data on web pages depend on unique choice of interest in extracting the record. This algorithm extracts data from heterogeneous web pages. It is insensitive to modifications in web-page format which enable to detect false positive rate in associating the attributes of records with their respective values. Experiments show that this method achieves higher accuracy than existing methods in automated information extraction.

Keywords: Automatic data extraction, data record alignment, Structural-Semantic Entropy, False positive rate detection

1. Introduction

The world wide web is large repository of information on growing demand. It is very hard to query in unstructured data. This is due to abundance of pages in structured data which is generated dynamically from relational database. Extraction of structured data is very much feasible for complex queries in the web page over the data which integrate the data present in different web-sites. Hence it is the challenging issue in information extraction in automated fashion.

This paper focuses on the issues of Automatically extracting data records that are encoded in the query result pages generated by web databases. It does not involve any human input like manually generated rules or training sets. For instance, from a collection of pages in shopping extract item tuples where each tuple consists of name of the product, product type, the number of items, the list-price and other attributes. Traditional data extraction from web pages uses the concept called wrappers" or "extractors". It extracts the contents of the web pages based on the knowledge of their formats which was developed manually in early time. In other words, programmers have to observe the extraction rules in person and write wrappers for each web site. These processes require numerous manual coding and debugging. Since even small modification at the web site may prevent the proper functionality of wrappers and the layout of web pages is often subject to change. It is most expensive and inefficient to maintain those wrappers. There are three stages to extract objects from a Web page. It includes record extraction, attribute alignment and attribute labeling[5]. For any given Web page, the first step identifies a web record. i.e., A set of HTML regions, each of which represents an individual object (e.g., a product). The second step is to extract object attributes (e.g., product names, prices, product type and owner of the product) from a set of Web records. Those attributes from heterogeneous Web records are aligned resulting in spreadsheet-like data. The final step is the optional task of deducing aligned attributes and giving appropriate labels.

The rest of the paper is organized as follows: Section 2 provides background information from related works in data extraction methods. Section 3 presents a notion of structural-semantic entropy, which could be used to recognize the data of interest in web pages.Section 4 describes the experiments, results and model obtained; and finally, conclusions and future works are outlined in Section 5.

2. Background

Extracting structured data from HTML pages has been is based on wrapper induction[2]. It utilizes manually labeled data to learn data extraction rules. Such semi-automatic methods are not scalable enough for extraction of data on the scale of the Web. To address this limitation, more fully automatic methods have been studied recently. Fully automatic methods address two types of problems: (1) extraction of a set of data records from a single page and (2) extraction of underlying templates from multiple pages. The former does not assume the availability of multiple instance pages containing similar data records [10]. Techniques that address record extraction from a single page can be categorized into the following approaches which evolved in this order: (a) early work based on heuristics, (b) mining repetitive patterns and (c) similarity-based extraction. Some data may contain embedded tags which may confuse the wrapper generators making them even less reliable. To overcome these shortcomings methods such as ViPER and ViNTs make use of additional information in the query result pages. ViPER uses both visual data value similarity features and the HTML tag structure to first identify and rank potential repetitive patterns. Then matching subsequences are aligned with global matching information while ViPER suffers from poor results for nested structured data. Using both visual and tag features ViNTs learns a wrapper from a set of training pages from a website. It first utilizes 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. Both visual and nonvisual features are used to weight the relevance of different extraction rules. Several result pages, each of which must contain at least four QRRs and one no-result page are required to build a wrapper. ViNTs has several drawbacks. First, if the data records are distributed over multiple data regions only the major data region is reported. Second, it requires users to collect the training pages from the website including the no result page which may not exist for many web databases because they respond with records that are close to the query if no record matches the query exactly. Third, the pre-learned wrapper usually fails when the format of the query result page changes. Hence it is necessary for ViNTs to monitor format changes to the query result pages which are most difficult problem. In addition to the above techniques, there is a another data extraction method called as Combined Tag Value Similarity(CTVS)[1], which extracts QRRs from a query result page in an automated fashion. CTVS employs two steps for this task. The first step identifies and fragments the QRRs. The existing technique is enhanced by allowing the QRRs to be non-contiguous in the data region. The second step is to align the data values among the QRRs. A novel alignment method consists of three consecutive steps: pair wise alignment, holistic alignment and nested structure processing. CTVS mainly focuses on the issue of extracting data records that are determined in the query result pages automatically generated by web databases [1]. In general, a query result page contains actual data, but also other information such as navigational panels, advertisements, comments, information about hosting sites and so on. The aim of web database data extraction is to remove any insignificant information from the query result page, extract the query result records and align the extracted QRRs into organised table in which data values belonging to the same attribute are placed into corresponding table column.

Combining Tag and Value Similarity (CTVS) is the two-step method to extract the QRRs from a query result page p.

1. Record extraction identifies the QRRs in *p* and involves two substeps: data region identification and the actual segmentation step.
   - Data region identification identifies the non contiguous QRRs that have the same parents belong to their tag similarities.
   - Segmentation step combine different data regions that contain the QRRs (with or without the same parent) into a single data region.
2. Record alignment line up the extracted QRRs into organised table in which data values belonging to the same attribute are placed into corresponding table column by pair wise and then holistically based on tag structure similarity and data value similarity.
   - The data values within the same attribute have the same data type and similar data values because they are the result for the same query in pair wise alignment.
   - After all pairs of records are aligned in pair wise alignment, holistic alignment is performed by viewing the pair wise alignment result as a graph and finding the connected components from the graph. Hence CTVS uses both tag and data value similarity information to improve nested structure processing accuracy.

In query result page, a tag tree is constructed during Tag Tree Construction phase rooted in the <HTML> tag. Each node represents a tag in the HTML page and its children are tags enclosed inside it. Each internal node n of the tag tree has a tag string ts_n. It includes the tags of n and all tags of n's descendants, and a tag path tp_n, from the root to n. Then, the Data Region Identification module identifies all possible data regions starting from the root node in top down manner. According to the tag patterns the Record Segmentation fragment the identified data regions into data records. Data Region combines the data regions containing correlated records. At last, the Query Result Section Identification selects one of the merged data regions as the one that contains the QRRs.

But the existing CTVS data extraction method still suffers from some limitations as listed below

- It is in need of at least two QRRs in the query result page.
- The start node in a data region is considered as optional attribute which is treated as auxiliary information.
- CTVS based on tag structures is used to find out data values.
- CTVS does not be used where multiple data values from more than one attribute are clustered inside one leaf node of the tag tree and also where one data value of a single attribute spans multiple leaf nodes.

3. Proposed Extraction Algorithm

The proposed information extraction algorithm locates and extracts the data of interest from web pages across different sites [14]. Our approach is different from the previous method in following aspects:

- Our algorithm are designed for the record-level extraction tasks that discover record boundaries, divide them into separate attributes and associate these attributes with their respective values automatically. This algorithm does not require any interaction with the users during the extraction process.
- It works without the requirements that the web pages need to share the similar template or multiple records need to occur in a single web page. The algorithm can treat a single web page containing only one record. If a set of keywords used to describe the data of interest is collected, the extraction is fully automated and it is easy to move from one application domain to another.
- It work properly when the format features of the source pages change, thus it is completely insensitive to changes in web-page format.

An automated information extraction algorithm that can extract the relevant attribute-value pairs from product descriptions across different sites. The proposed method known as structural-semantic entropy is used to locate the data of interest on web pages which measures the density of occurrence of relevant information on the DOM tree representation of web pages. Our approach is automatic and insensitive to changes in web-page format.

3.1 Structural-Semantic Entropy

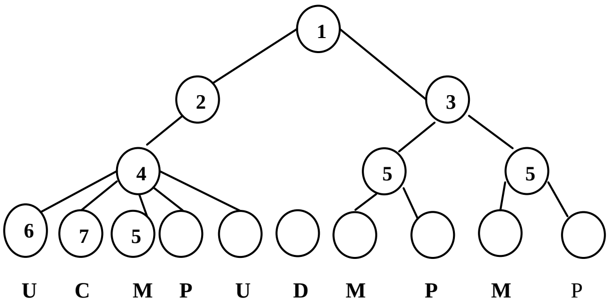
The concept of structural-semantic entropy is used to identify and locate the data-rich nodes. We define the structural-semantic entropy of a node in a DOM tree in terms of the semantic roles of its descendant leaf nodes. The leaf nodes are all annotated with their corresponding semantic roles, i.e., attributes of a product. The annotation process can be accomplished by identifying metadata labels in the web pages, and a leaf node that does not belong to any of the semantic roles of interest is assigned to be unidentified. For each attribute, a set of keywords used to indicate this attribute is collected previously. An important thing to be notice is, we have to collect some of these keywords, but not all of them, to run the algorithm, and this task can be done easily without a high level of expertise. The more keywords we collected, the better the results will be.

Definition : The structural-semantic entropy H(N) of a node N in the DOM tree representation of a web page can be defined as

$$H(N) = -\sum_{i=1}^{m} p_i \log(p_i)$$

Where p_i is the proportion of descendant leaf nodes belonging to semantic role *i* of the node *N* and conventionally the base of the logarithm is 2.

Entropy is a measure of disorder, or uncertainty in the system. More entropy means more possible variation and hence greater capacity for storing and transmitting information. Thus the node contains higher structural-semantic entropy possess higher data-rich region.



U: Unidentified
C: Product Code
M: Manufacturer
P : Price
D: Description

Structural- Sematic Entropy
1: 1.9360125 2: 2.256982
3: 1.0000000 4: 2.256982
5: 1.0000000 6: 0.000000
7: 0.0000000 8: 0.000000

Fig 1: Sample DOM tree and the structural semantic entropies of the nodes in the DOM tree

Figure 1 shows the simplified DOM tree for the right side of HTML page. Each leaf node has been annotated with its semantic role and there are five different kinds of semantic roles: product code, manufacturer, price, description, and unidentified. The unidentified nodes do not belong to any of the first four semantic roles, therefore they are considered as noises.

The numbers at the bottom right of Figure 1 show the values of structural-semantic entropy for some representative nodes in the DOM tree. The following rules can be observed:

- The structural-semantic entropies of all leaf nodes are zero.
- The higher structural-semantic entropy a node has, the more likely the node is a data-rich node.
- When a node and its child node have the same structural semantic entropy and one of them needs to be selected as the data-rich node, we always choose the child node.

- The structural-semantic entropy of the lowest common parent node (called list node) of the sibling subtrees forming a list is equal to or close to 1, where the root nodes of these sibling subtrees have the same or similar values of the structural semantic entropy.

For example, since node 2 and node 4 in the DOM tree of Figure 1 have the same entropy, and node 2 is parent of node 4, node 4 will be selected as a data-rich node. Node 3 is a list node because its entropy is equal to 1, and its five child nodes have the same value of entropy. A link offer node is the parent node of the sibling subtrees forming a "link offer", and it can be seen as a special kind of list node, where its child nodes have lower structural-semantic entropy than the data-rich node. Node 3 is also a representative link offer node.

Algorithm 1: *DE-SSE* (Extraction of data from web pages based on Structural-Semantic Entropy

```
Input:  P:   a web page
        R:   a set of semantic roles for a given
             domain, each of which has a set of
             keywords Kr used to annotate the
             leaf nodes with the semantic role r
        Hd:  a threshold used to identify the data-
             rich nodes
        Hl:  a threshold used to identify the list
             nodes
Output: V:   a set of attribute-value pairs of
             records
Begin
deletes the bad HTML tags and syntactical errors in P and turns the body of P into a DOM tree, T.
discard HTML attributes and representation tags, such as b, i and font, from T
for each leaf node i in T do
    if the content of i matches any keyword in Kr then
        annotate i with the semantic role r
    if the content of i does not match any keyword then
        annotate i with the unidentified role
    if i is annotated with d ($d > 1$) semantic roles then
        separate i into d nodes, and annotate d nodes with their corresponding semantic roles
traverse T in a breadth-first way, and sort all non-leaf nodes of T in the reverse order of the traversal sequence
for each non-leaf node j in T do
    calculate the structural-semantic entropy ej for j
    if $ej >= Hd$ and j has a greater structural-semantic entropy than all its descendant nodes then
        j is a data-rich node, and makes all its descendant nodes non data-rich node.
    if $Hl <= ej < Hd$ and j is the common parent node of the sibling nodes that have the same nonzero values of structural-semantic entropy then
        j is a list node
        if any structural-semantic entropy of the sibling nodes is less than Hd then
            j is a link offer node
for each data-rich node m do
    for each leaf node n of m that is annotated with a semantic role do
        extract a value for the semantic role (if the regular expression for matching value is defined, the value should be tested) and associate the value with the corresponding attribute
    insert the attribute-value pairs of a record into V
return V
end { DE-SSE }
```

3.2 Algorithm Description

The proposed algorithm navigates on webpage in bottom-up manner based on DOM tree representation. The structural-semantic entropy discovers the data-rich nodes and list nodes from the web page by *DE-SSE* algorithm. This algorithm identifies data-rich regions and extracts the attribute-value pairs from those regions in automated manner. For every attribute of a product, a regular expression is constructed to equivalent the keywords in order to deduce the attribute values. So the leaf nodes can be interpreted with their semantic entropy measures. For example, the metadata is used to discover the nodes with the semantic role "Item" and check the feasible values for the role "Item". The scheduled synonyms were employed to find out a regular expression for recognizing the "Item" nodes. These synonyms are accumulated increasingly. When these collected synonyms are increased in number, the precision and recall of the algorithm will be in higher rate. The nature of a node depends on the states of its child nodes. This is due to structural-semantic entropies of the nodes in a DOM tree representation. When a node and its parent have the similar entropy build the child node as a data-rich node except the ancestor of the node has greater structural-semantic entropy. The proposed algorithm recognizes the record boundary repeatedly from each data rich node containing a record. The content of the next text-node of the node that interpreted with a semantic role is usually extracted as the value of the semantic attributes.

For some types of attributes such as price and time, regular expressions are constructed to increase the precision. This is to ensure that whether the strings are valid values or not for those attributes. If the extracted string is not valid for the attribute, the content of the next-next node will be extracted until meet up the node interpreted with a different semantic role. There are many circumstances in which the title of a record is not explicitly associated with a string. However, they can be extracted from the first leaf child of the data-rich nodes or the previous leaf of that node. On the other hand the titles frequently arise in the same relative position with respect to the data-rich nodes. Our algorithm is much more efficient in data extraction based on the requirements that the web pages share the similar template as well as dissimilar template. On the other hand it handles multiple records occur in homogeneous as well as heterogeneous web page. This algorithm extracts the interested data region from one web page containing only one record. Once a set of keywords known as interested data region are collected, the extraction process is automated to make it simple for non-experts unless those keywords is changed from one application to another. Thus Wrappers created using our approach are inherently adaptable and supple when the format of the source pages changes even. This is also beneficial for web pages from distinct sources belonging to the same application domain.

4. Results And Discussion

4.1 Performance Metrics

In general, A**ccuracy** is the degree of closeness of measurements of a quantity to the true value. The **Precision (**repeatability) is the degree to which repeated measurements under unchanged conditions show the same results. Thus increase in precision results as reduction in recall. This arise ambiguity when they are used to extract multi-attribute data.

$$\text{Precision} = \frac{\text{True positives}}{\text{True positives} + \text{False positives}}$$

In our context**, Precision** is the fraction of retrieved instances that are relevant while **Recall** is the fraction of relevant instances that are retrieved. Thus increase in recall results in reduction in precision:

$$(\text{Recall}) = \frac{\text{True positive}}{\text{ground truth}}$$

The experimental result for structural semantic entropy is compared with CTVS and ViPER. The performance result of ViPER is limited whereas the proposed shown to perform very accurate data extraction. Data set 1 (Amazon.com) contains 80 websites. Among 80 websites 30 websites return relational records such as jobs and entertainment records and 60 return documents. For each of the 80 websites, 10 queries are submitted and the first 10 result pages are collected manually. By submitting nonexistent term as a query to the website no-result page is also collected for it. For each website, its no-result page and five randomly selected result pages from the 10 result pages are used to build a wrapper to extract the QRRs from the remaining five result pages Data set 2 (E-COMM) contains 80 E-commerce deep websites in six popular domains: Academics, restaurant, employment, fun, Music Record and vehicle. Each domain contains 20-35 websites. For each website, five result pages are created as training pages by submitting five queries and one test page for as a test page by submitting another query. Compared with data set 1, it is found that the QRRs in E-COMM have more complex structures since they usually contain more nested levels and more optional attributes in the page HTML tag tree which reduces the data extraction precision.

Table 1: Data Extraction Performance

Applications	**E-COMMERCE**			**Amazon.com**		
#QRRS	6900			970		
Method	ViPER	CTVS	Entropy	ViPER	CTVS	Entropy
#pairs	1800	6500	1200	540	870	860
#Correct QRRS	1745	6400	1190	530	800	810
Precision(%)	93.2	92.9	94.6	90.26	94.3	99.47
Recall(%)	87.8	90.50	95.5	93.40	96.2	98.93
Page-level Precision(%)	70.2	80.2	90.3	92.1	92.1	93.2

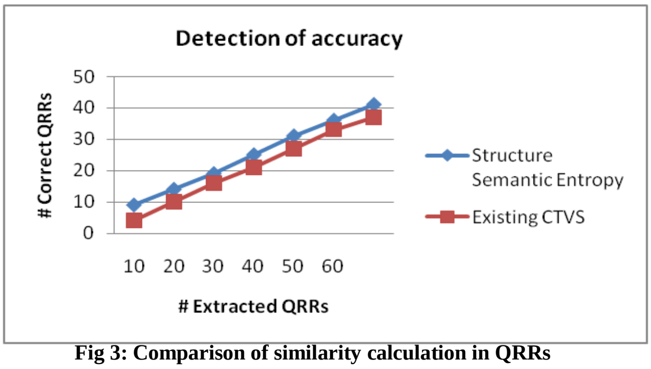


Fig 3: Comparison of similarity calculation in QRRs

5. Conclusion

The Existing Data Extraction Method (CTVS) allows the Query Result Records in a data region to be non-contiguous as well as aligns the data values among the QRRs. Although it has been shown to be an accurate data extraction method it does not figure out the case where multiple data values from more than one attribute are clustered inside one leaf node of the tag tree and data value of a single attribute spans multiple leaf nodes. The proposed structural-semantic entropy is calculated for each node in a DOM tree. It focus on recognizing data-rich regions and find the lowest common parent nodes of the sibling subtrees forming the records in the DOM tree representation of a web page with the help of a set of domain keywords. The future work may be extended to extract the data from web pages based on the design issues such as memory consumption, computational overhead, storage, fast processing etc. The current algorithm requires that the entropy should be calculated for every non-leaf node of a DOM tree. One of the possible approach is to find rules to terminate the calculation before the entropies of all nodes are calculated in a bottom-up way. On the other hand accelerate data extraction for the pages during the process of crawling a web site.

References

[1] Weifeng Su, Jiying Wang, Frederick H. Lochovsky ,"Combining Tag and Value Similarity for Data Extraction and Alignment"', IEEE Transactions on Knowledge and Data Engineering, Vol. 24, No.7,pp. 1186- 1200, July 2012.

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

[3] D. Buttler, L. Liu, and C. Pu, "A Fully Automated Object Extraction System for the World Wide Web," Proc. 21st Int'l Conf. Distributed Computing Systems, pp. 361-370, 2001.

[4] D.Pramod Krishna,T. Swarna Latha and T.Rajasekhar Reddy, "Extracting Web Data Based On Partial Tree Alignment Using Fivatech", International Journal of Advanced Research in Computer Science and Software Engineering, Vol.2,No.3, pp. 369-373,March 2012.

[5] Gengxin Miao,Junichi Tatemura, Wang-Pin Hsiung,Arsany Sawires, Louise and E. Moser, "Extracting Data Records from the web using Tag Path Clustering." Proc. WWW 2009 MADRID 2009, pp. 981-990,2009.

[6] Yanhong Zhai and Bing Liu, "Web Data Extraction Based on Partial Tree Alignment", Proc Int'l World Wide Web Conference Committee (IW3C2), 2005.

[7] K.C.-C. Chang, B. He, C. Li, M. Patel, and Z. Zhang, "Structured Databases on the Web: Observations and Implications," SIGMOD Record, vol. 33, no. 3, pp. 61-70, 2004.

[8] C.H. Chang and S.C. Lui, "IEPAD: Information Extraction Based on Pattern Discovery," Proc. 10th World Wide Web Conf., pp. 681-688, 2001.

[9] Álvarez, M., Pan, A., Raposo, J., Bellas, F., and Cacheda, F. Extracting lists of data records from semi-structured web pages. *Data & Knowledge Engineering*, 64, pp. 491-509, 2008.

[10] Embley, D. W., Campbell, D. M., Jiang, Y. S., Liddle, S. W., Lonsdale, D. W., Ng, Y.-K., and Smith, R. D. Conceptual model based data extraction from multiple-record Web pages. *Data & Knowledge Engineering*, 31, pp. 227-251,1999.

[11] Etzioni, O., Cafarella, M., Downey, D., Popescu, A.-M.Shaked, T., Soderland, S., Weld, D. S., and Yates, A. Unsupervised named-entity extraction from the web: an experimental study. *Artificial Intelligence*, 165, pp. 91-134,2005.

[12] Vadrevu, S., Gelgi, F., and Davulcu, H. Information Extraction from web pages using presentation regularities and domain knowledge. In Proceedings of the World Wide Web (WWW'07). Springer, pp. 157-179,2007.

[13] Wong, T.-L., and Lam, W. An unsupervised method for joint information extraction and feature mining across different Web site. *Data & Knowledge Engineerin*g, 68, pp. 107-125,2009.

[14] Xiaoqing Zheng, Yiling Gu and Yinsheng Li",Data Extraction from Web Pages Based on Structural-Semantic Entropy", pp. 93-102,April 16–20, 2012

[15] Cohen, W. W., Hurst, M., and Jensen, L. S. A flexible learning system for wrapping tables and lists in HTML Documents. In *Proceedings of the World Wide Web (WWW'02)*. pp. 232-241,2002.

[16] Dongdong Hu and Xiaofeng Meng, "Automatic Data Extraction from Data-rich Web Pages", In Proceedings of DASFAA'05',10th international conference on Database Systems for Advanced Applications,pp. 828-839,2005.