Volume 1 | No. 2 | July – December 2017

Schema Integration of Web Tables (SIWeT)

Nayyer Masood[1*], Amna Bibi[2], Muhammad Arshad Islam[1]

Abstract:

Schema integration has been mainly applied in database environment whether schemas to be integrated belonged to a single organization or multiple ones. Schema extraction is a relatively new area where schema is extracted from a web table. The extracted schema is not as much concretely defined as in a typical database environment. The work in this paper brings two areas together where extracted schemas from multiple web tables are integrated to form a global schema. The data are also extracted from web tables and placed into global table. This creates a large repository of data of the same domain extracted dynamically from websites which is then available for different types of ad-hoc queries. This work also imposes challenges on schema integration to be studied in the context of schema extraction and other way round.

Keywords*: S*chema Extraction, Schema Iintegration, Semantic Heterogeneities, Web Tables.

1. Introduction

Web is a tremendous source of huge volume of data which is in structured, semi-structured and unstructured formats. Structured data include lists and tables. A table is a collection of rows containing data in one or multiple columns. Each column represents an attribute. Web tables are a simple, meaningful, effective and popular way of representing data on web. Data from these sources can be accessed either using search engines or navigating through the web pages. However, structures (or schemas) of these web tables are not available, so they cannot be queried efficiently; hence their data mostly remain inaccessible to the users.

Extracting data from these tables and storing it in a database could be very useful for many value added services, like in business and social web domains. For example, in business domain, data extraction techniques help in reducing time and manpower and increase the business efficiency. This data can help the analysts and manager to revise or change the business strategies and plans. Context-aware advertising, customer care, comparative shopping, Meta query, opinion mining and database building are the major applications of web data extraction techniques [1].

Schema extraction is the process of extracting schema from the structured data (e.g. tables, spreadsheets) on the web. The process is followed by fetching the also data from these web tables. The extracted schema is used to create tables in a database, which are populated with the extracted data. The database tables can then be used for better and efficient querying.

Many sites on the web can be found that belong to the same domain. For example, sites from the banking domain, education, entertainment etc. We can apply schema extraction on multiple sites of the same domain and can store tables extracted from these sites at a single place for efficient querying. However, even in this case the query will be applied to individual tables if data are to be accessed from multiple sites. It will be more beneficial if schema integration could be applied on schemas extracted from the multiple sites of the same domain. Schema integration is the process of merging/combining same or similar data items to build a canonical schema. For example,

[1] Department of Computer Science, Capital University of Science & Technology, Islamabad, Pakistan

[*] Corresponding Author: nayyer@cust.edu.pk

[2] Department of Computer Science, Virtual University Rawalpindi, Pakistan

every university website shows its faculty information generally including faculty member's name, his/her designation, higher degree and research interests etc. Many of the university websites show this data in the form of web tables due to ease in creating, visualizing and understanding of tables. If schema extraction is applied to multiple universities' web sites to fetch the schema and data and store them in database tables; further schema integration can be on these database tables to get a canonical schema. This will give us a single schema/table containing data of faculty members belonging to different universities. We can then apply advanced/efficient queries to extract required information about faculty members of different universities.

Structure of this paper is as follows: section 2 presents the review of the related literature encompassing both schema extraction and schema integration. Section 3 presents proposed approach comprising three phases. Finally, section 4 concludes the paper.

2. Literature Review

The review of related literature falls into two major categories: schema extraction and schema integration. Both of them are discussed in the following:

2.1. Schema Extraction

Several approaches are available for schema extraction that can broadly be categorized as manual, wrapper induction and automatic extraction. In wrapper induction [2], [3], firstly the pages/data records are labeled manually and a set of extraction rules is deducted. These rules are then used to extract data from similar pages. This technique still involves manual effort. Automatic method [4], [6] finds patterns or grammars from similar pages containing similar data records and uses these patterns to extract data from new pages. The pages to extract the patterns are provided manually or by some other system.

The approach of Zhai & Liu [6] provides an automated system to extract data from web and put it in a database. Web pages are constructed using HTML tags. The <table> tag is used to represent table on web. The <tr> tag is used to insert the rows and <td> tag inserts the data in a particular cell of that row.

The proposed approach firstly identifies the data records. For this purpose, visual information is used to construct a tag tree which is constructed by following the nested structure of HTML code. Second task is to align and extract data from the identified data records using partial tree alignment technique. Tree edit distance is used to identify the data records. Trees are matched with each other node by node. Trees are aligned by gradually growing a seed tree Ts. The tree with the maximum records is chosen as the starting seed tree. Then for each node n_i in T*i* a matching node n_s is found in T_s. When a matching node is found in T_s, a link from n_i to n_s is created. If no match is found for n_i then seed tree is expanded by inserting this node into it. Data item nodes are not used during the matching. The data item in the matched nodes children is inserted into one column in database table.

Adelfio & Semat [4] proposed a conditional random field (CRF) based classification technique with the combination of logarithmic binning. Each web table contains different types of rows, like caption, heading, empty and data rows etc. Each row has been classified based upon the row features which include formatting, layout, style and values of the cell and then all these features are combined using binning to construct row features. In next step, logarithmic binning method is applied in which individual cell attributes are used collectively to encode row features. For each possible row feature a bin is formed and each bin is assigned a value which represents its feature. After row features extraction, row labels are assigned to each row based on CRF. CRF is trained with human classified rows. After training the CRF is used to label huge volume of data. The output of the CRF is a sequence of row labels like "TNNHGDDDAGDDDABN". This output helps in extracting schema of the relational table. Column names are decided based upon the header row(s), data type is determined by the type frequency within the data rows of each column, additional attributes can be determined by the group header rows and data rows are determined by the data records.

George, David and Sharad [7] introduced a technique to covert the web tables to relational tables. This is the first end to end approach which produces an access compatible canonical table. The HTML table is converted into an excel table and from excel table its CSV file is generated. Table is segmented based upon the indexing property rather on appearance features. To segment the table minimum indexing point (MIP) and four critical cells CC1, CC2, CC3 and CC4 are calculated. CC1 and CC2 determine the stub headers; CC3 and CC4 indicate the data regions. MIP (CC2) is determined by searching from the cell A1 for unique columns and row header rows. The categories can be extracted by comparing the number of unique elements in the cross-product of a pair or header rows with the length of the header. From the category extraction output, canonical table is generated. This table can be used to query the data.

The technique proposed by Purnamasari, Wicaksana, Harmanto and Banowosari in [8] first finds the area of the table and then extracts data from it. First of all table is detected and then property (title) portion of the table is detected before extracting data from it. The technique is divided into three steps and algorithm for each step is formulated. In first step, number of rows and columns are calculated by counting the <tr>... </tr> tags in <table> tag and the <td>…</td> tags in each <tr> tag. The algorithm also checks the colspan attribute in the <td> tag. It adds the value of colsapn in the column count. In second algorithm the property of the table is detected. Generally the first row of the table contains the headings of the columns. The algorithm checks for the row span attribute in each <td> tag in <tr> tag of table to calculate the length of the property of the table. Third algorithm actually extracts the data from the table. It takes the value of the rowspan returned from the second algorithm to extract the heading of the columns. While reading the data in <td> tag of <tr> tag, it checks the value of colspan. If its value is greater than 1, it concatenates the content in this cell with the columns below it. After reading the header rows, it reads the cells row by row.

2.2. Schema Integration

It is the process that takes two or more schemas as input (called source or member schemas) and merges them into a single/canonical one. Other terms used for schema integration are database integration, data integration, database interoperability, etc. The most critical step in schema integration is schema matching in which two schemas are compared with each other to identify the elements modeling same or similar concepts. Once identified, the similar elements are merged into a common one in the schema merging phase, which is a relatively straightforward task. The main problem in schema matching is identification and resolution of semantic heterogeneities. A semantic heterogeneity reflects a situation when same or similar concept is modeled differently in two schemas. These differences arise mainly due to differences in the context of organization, popularity of using acronyms in defining schemas, idiosyncratic abbreviations and models of the same domain [11]. The schema integration approaches can be broadly categorized into two; schema based and instance based.

Schema based integration approaches exploit the information in the schema, like, name, data type and different types of constraints, to identify semantically similar schema elements. These techniques have been further classified as element-level and structure-level in [13]. Element-level schema matching approaches compare the entity types from different schemas in isolation without considering their links/relationships with other entity types. These approaches mainly include string-based techniques [14], [15] that use matchers like prefix, suffix, edit-distance and N-gram; NLP-based techniques [16, 17] that apply natural language processing techniques, like tokenization, lemmatization and elimination on the names of the entity types and then apply some matching technique; constraint-based techniques [18] where constraints from schema are used to compare the schema elements, like data type, integrity constraints etc. Structure-level approaches mainly cover graph-based [19], taxonomy-based [20] and model-based [21]. A hybrid approach has been adopted in COMA++ [22], where a combination of matchers is applied on input schemas and the results are combined to establish final similarity between elements.

Table I. Comparison of Different SE Approaches

S #	Paper Reference	Schema Extraction	Fully Automated	File Format	Techniques	Data Domain	Multiple Sources
1	4	Yes	Yes	HTML, Table, Spreadsheet	Supervised	Different domain	No
2	6	No	Yes	HTML table	Tree based	Shopping data	No
3	7	No	Yes	HTML, table, Spreadsheet	Index Based	Statistical data	No
4	8	No	Yes	HTML tables	Programming	Not mentioned	No
5	9	No	Yes	HTML tables	Tree based	Different domains	No
6	10	No	Yes	HTML tables	Tree based	Different domains	No

The literature review of schema merging approaches reveals that most of the approaches strive to maximize the automated part of the process as performing SI completely manually is very time consuming and laborious task. Moreover, most critical part of SI process is handling semantic heterogeneities which exist across multiple schemas due to the fact that these schemas are built independently in certain contexts that are entirely different from each other even if they belong to same domain.

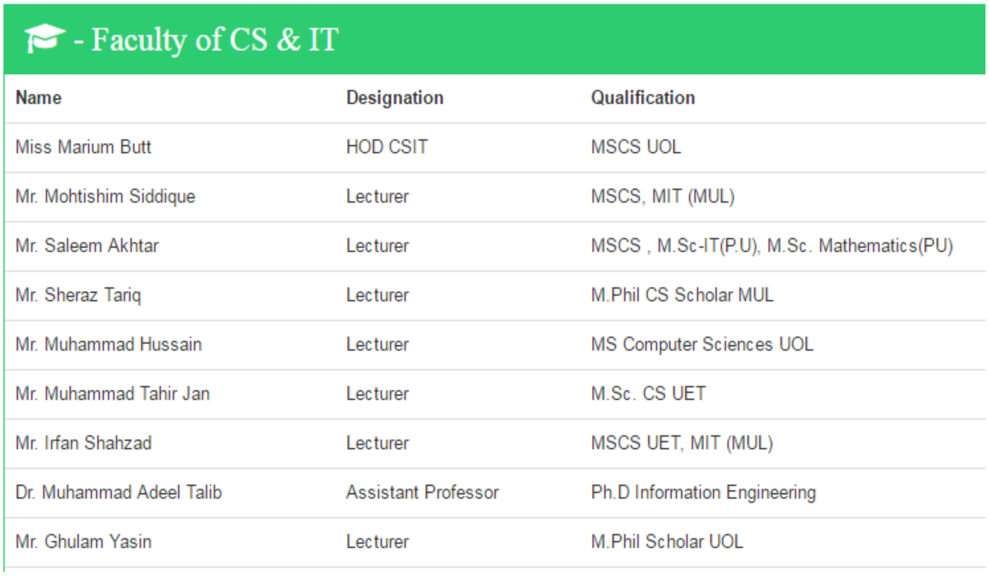
3. Proposed Approach

This article presents the novel idea of applying schema integration (SI) process on the schemas that have been extracted through schema extraction (SE) process from web tables of multiple web sites belonging to the same domain. To prove the concept, it is planned to test proposed approach on the domain of faculty members of computer science departments of different universities. However, the idea can be applied in any domain. The basic idea behind this approach is to access those websites where the data of faculty members have been placed in the form of tables, as shown in Fig. 1 below. Then, using the SE approach presented in [8], extract the schema and data from different websites. After that, different schema matching approaches will be applied to identify semantically similar elements among the elements extracted from different universities websites.

The semantically similar elements are merged with each other and the data are finally stored in a single table for further queries. The proposed approach comprises three major phases; preprocessing, schema extraction and schema integration. In the following, these three phases have been explained.

3.1. Preprocessing

Basic objective of this step is to provide neat and clean web table source to SE step so that an accurate schema could be extracted out of it. Neat and clean web page source means removing all unnecessary or irrelevant code or tags from the web page source that means any source or tags other than that contains the web table including table headings and data.



🎓 - Faculty of CS & IT

Name	Designation	Qualification
Miss Marium Butt	HOD CSIT	MSCS UOL
Mr. Mohtishim Siddique	Lecturer	MSCS, MIT (MUL)
Mr. Saleem Akhtar	Lecturer	MSCS , M.Sc-IT(P.U), M.Sc. Mathematics(PU)
Mr. Sheraz Tariq	Lecturer	M.Phil CS Scholar MUL
Mr. Muhammad Hussain	Lecturer	MS Computer Sciences UOL
Mr. Muhammad Tahir Jan	Lecturer	M.Sc. CS UET
Mr. Irfan Shahzad	Lecturer	MSCS UET, MIT (MUL)
Dr. Muhammad Adeel Talib	Assistant Professor	Ph.D Information Engineering
Mr. Ghulam Yasin	Lecturer	M.Phil Scholar UOL

Figure 1: An Example Web Table of Faculty Data

It is a critical and difficult task, as there is too much difference the way web tables are defined on different web sites.

In the first step of preprocessing phase, web sites of universities will be found manually that store the faculty data in the web table form and store that URL in a database table along with the other basic information about the university and department. This is an ongoing process and the database of university pages will keep on increasing. Once we have that data, this table will be handed over to a crawler which picks the URL of websites one by one and downloads the source code of the web pages and stores it in a text file. In the next step, clipping is performed and additional or irrelevant code/tags are removed and only the part contained within <table> and <\table> tags are left that contains the web table. This is going to be a bit tricky, as a web page generally contains many tables (for example, for formatting purpose) and out of those tables the one that presumably contains required data will be picked. One possible strategy in this regard can be, to pick the table that contains multiple rows and within each row there are multiple columns; this is also a requirement of SE approach [8] that has been selected in proposed approach. As an example, parts of HTML code of two web tables (after clipping) have been shown in Fig. 2 below. Both of these pages present the faculty data in the form of a table, but the variation in the coding can still be seen as the code in the right column contains a lot of formatting instructions whereas one on the left simply contains the data inside HTML tags. The SE approach that we have selected [8] assumes web table to be in a specific format (fig. 1). However, it is possible that a website does contain the <table>, <\table> tags but still is not in the required format. So one objective of preprocessing phase is to identify such pages and put them aside rather than passing those to next phase because the adopted approach will not be able to successfully extract schema out of such pages.

```
<table>
  <thead>
<th>Staff Name</th>
            <th>Staff
Designation</th>
            <th>Staff Image</th>
        </thead>
            <tr>
                  <td>Dr. Muhammad
Anwar-ul-Rehman Pasha</td>
                  <td>Professor</td>
                  <td></td>
            </tr>
            <tr>
<td>Mr. Abid Rafique</td>
<td>Assistant Professor</td>
            <td></td>
            </tr>
            <tr>
            <td>Dr. Muhammad Din
Choudhry</td>
                  <td>Assistant
Professor</td>
                  <td></td>
            </tr>
            …………..
                  <\table>
```

```
<table class="MsoTableGrid" border="1"
     cellspacing="0" cellpadding="0" width="100%"
     style="width: 100%; border-collapse: collapse;
     border: sube;">
     <tbody>
     <tr style="height: 18.4pt;">
 <td style="width: 10%; border: 1pt dotted
       windowtext; padding: 0in 5.4pt; height:
       18.4pt; background-color: #8db3e2;">
             <p class="MsoNoSpacing" style="text-
align:
                  center;"><strong><span style="font-
size:
                  9 pt; font-family: arial, sans-serif;">
 S.No<o:p></o:p></span></strong> </p></td>
   <td style="width: 56%; border-style: dotted dotted
    dotted none; border-top-color: windowtext;
    border-right-color:  windowtext; border-bottom-
    color: windowtext; border-top-width: 1pt; border-
    right-width: 1pt; border-bottom-width: 1pt;
    padding: 0in 5.4pt; height: 18.4pt; background-
    color: #8db3e2;">
    <p class="MsoNoSpacing" style="text-align:
        center;"><strong><span style="font-size: 9pt;
        font-family: arial, sans- serif;">
        Name<o:p></o:p></span></strong></p></td>
    ……………
</table>
```

Figure 2: Sample Code for Two Different Web Tables

3.2. Schema Extraction

There are many approaches proposed for SE in literature, but we have selected one proposed in [8] because it is quite recent, simple to understand and implement, moreover, it performs comparatively well on the web pages that are in the specific format assumed by the approach. There are many extensions possible in this approach, but our main objective is to apply SE for integration purposes, so we are using approach of [8] as such rather than suggesting any enhancement.

The SE approach that we have selected assumes that the web table is contained inside <table>, <\table> tags, and there are multiple rows between <table> <\table> tags. The first row contains the headings of the columns and remaining rows contain the data. One special feature of the approach is that it can manage the situations where header is spanned upon multiple rows. In this paper, we are not discussing SE approach of [8] in detail; interested readers can refer the actual paper. The preprocessing phase has already placed the clipped web table code in a text file. In this phase, SE will be applied to all web tables stored in text files and from each file the first row is separated as header row and remaining as data rows. Inside header rows there are different column names that are separately stored in an array. The n column names in the header row of the web table are stored in the first n elements of n array. Remaining rows of the web table are assumed to contain data. So data values from each row are stored in the next n elements of array. This process continues till all the rows of the web tables have been processed. At the end of this process, we have an array in which first n elements contain the names of the columns and each next set of n array components contain attribute values.

3.3. Schema Integration

Schema Integration (SI) is the third phase of our approach (SIWeT) where the extracted schemas in the previous phase will be merged to form a global schema. As mentioned earlier, semantic heterogeneities is the major problem faced in SI. This problem is further amplified when SI is implemented on extracted schemas where we do not have concretely defined schemas rather extracted from a web page by a semi-automated tool. Such schemas may have certain errors that may not exist in properly defined database schemas. Like, extracted schemas may have inappropriate or invalid data type assigned to an attribute because data types are assigned to attributes on the basis of evaluation of data, for example, a web table may contain date in number format like '021002' representing October 02, 2012, but SE approach may assign it numeric seeing all numeric data. There can be other such issues that make SI in extracted schema environment more complex as compared to a traditional database environment.

The SI approach in our SIWeT comprises applying multiple schema matchers on extracted schemas and then combining the output of these matchers. First of all, we define a global table for faculty members. This table is defined with maximum possible attributes that can be relevant to a faculty member. The SI task then becomes finding the most similar attribute in this global table for each of the attribute in every extracted schema. In order to find similarity between attributes, SIWeT builds taxonomy of similar terms using existing resources, like WordNet. In addition to this taxonomy, N-gram and edit-distance matchers are also applied. These matchers return the score of similarity between different terms. There scores are averaged and the pair having maximum similarity are considered as similar to each other. When corresponding attributes of extracted schemas have been found within the global table, then the data from web table will be inserted into the global table under the attributes found similar to the attributes of the extracted table, along with two additional attributes mentioning the university Id and department name. This process will be applied to all the extracted web tables from different universities and we will have a global table containing data containing data from many universities at a single place.

We plan to build a web application that lets the users query this table using simple keywords or SQL queries. This will be a great source of information for researchers and other interested users to find the relevant data.

4. Conclusion

In this paper, we have proposed application of schema integration approaches on the schemas extracted from web tables; so this work is basically a merger of two research areas, that is, SE and SI. This merger extends both areas as SE has been mostly applied on a single site in literature, whereas we are applying it on multiple sites. Our approach also extends SI research as the process has been mainly applied in database environment where properly defined database schemas are available defined through DBMSs, whereas we are applying it on extracted schemas. This will help to establish new dimensions in SI.

In future, we plan to implement our approach in real environment by extracting data from large number of web tables and merging them into a single table. The SE approach that we have adopted works on web tables in a specific format, there are many other formats of the web tables on which this approach cannot be applied. The SE approach can be extended to handle other web table formats. There are many web tables that store multiple attributes in a single column; we need to evaluate the data extracted from one column to identify relevant attributes.

REFERENCES

[1] Ferrara, E., De Meo, P., Fiumara, G., Baumgartner, R. (2014). Web data extraction, applications and techniques: A survey. *Knowledge-based systems*, 70, 301-323.

[2] Cohen, W. W., Hurst, M., Jensen, L. S. (2002). A flexible learning system for wrapping tables and lists in HTML documents. *In Proceedings of the 11th international conference on World Wide Web* (pp. 232-241).

[3] Kushmerick, N. (2000). Wrapper induction: Efficiency and expressiveness. *Artificial Intelligence*, 118(1-2), 15-68.

[4] Adelfio, M. D., Samet, H. (2013). Schema extraction for tabular data on the web. *In Proceedings of the VLDB Endowment*, 6(6), (pp 421-432).

[5] Zeeshanuddin, S. (2011) A library of schema matching algorithms for dataspace management systems.

[6] Zhai, Y., Liu, B. (2005). Web data extraction based on partial tree alignment. *In Proceedings of the 14th international conference on World Wide Web* (pp. 76-85).

[7] Nagy, G., Embley, D. W., Seth, S. (2014). End-to-end conversion of HTML tables for populating a relational database. *In 11th IEEE International Workshop on Document Analysis Systems (DAS)* (pp. 222-226).

[8] Purnamasari, D., Wicaksana, I. W. S., Harmanto, S., Banowosari, L. Y. (2015). HTML table wrapper based on table components. *International Journal of Computer Applications in Technology, 52(4),* 237-243.

[9] Lerman, K., Knoblock, C., Minton, S. (2001). Automatic data extraction from lists and tables in web sources. *In IJCAI-2001 Workshop on Adaptive Text Extraction and Mining* (Vol. 98).

[10] Gultom, R. A., Sari, R. F., Budiardjo, B. (2011). Proposing the new Algorithm and Technique Development for Integrating Web Table Extraction and Building a Mashup. *Journal of Computer Science*, *7(2)*, 129-136.

[11] Lukyanenko, R., Evermann, J. (2011). A Survey of Cognitive Theories to Support Data Integration. *In Proceedings of the Seventeenth Americas Conference on Information Systems,* All Submissions. Paper 30.

[12] Evermann, J. (2009) Theories of meaning in schema matching: An exploratory study. *Information Systems 34(1)*, 28-44.

[13] Shvaiko, P., Jérôme, E. (2005). A survey of schema-based matching approaches. *Journal on Data Semantics IV,* (pp. 146-171). Berlin, Heidelberg: Springer-Verlag.

[14] W. Cohen, P. Ravikumar, and S. Fienberg (2003). A comparison of string metrics for matching names and records. *In Proceedings of the workshop on Data Cleaning and Object Consolidation at the International Conference on Knowledge Discovery and Data Mining (KDD)* (Vol. 3, pp. 73-78).

[15] H. H. Do and E. Rahm (2001). COMA - a system for flexible combination of schema matching approaches. *In Proceedings of the Very Large Data Bases Conference (VLDB)* (pp. 610–621).

[16] F. Giunchiglia, P. Shvaiko, and M. Yatskevich (2004). S-Match: an algorithm and an implementation of semantic matching. In Proceedings of the European Semantic Web Symposium (ESWS), (pp 61–75).

[17] J. Madhavan, P. Bernstein, and E. Rahm (2001). Generic schema matching with Cupid. In Proceedings of the Very Large Data Bases Conference (VLDB), (pp 49–58).

[18] P. Valtchev and J. Euzenat (1997). Dissimilarity measure for collections of objects and values. Lecture Notes in Computer Science, 1280, (pp 259–272).

[19] D. Shasha, J. T. L. Wang, and R. Giugno (2002). Algorithmics and applications of tree and graph searching. In Proceedings of the Symposium on Principles of Database Systems (PODS), (pp 39–52).

[20] N. Noy and M. Musen (2001). Anchor-PROMPT: using non-local context for semantic matching. In Proceedings of the workshop on Ontologies and Information Sharing at the International Joint Conference on Artificial Intelligence (IJCAI), (pp 63–70).

[21] P. Bouquet, L. Serafini, and S. Zanobini (2003). Semantic coordination: A new approach and an application. In Proceedings of the International Semantic Web Conference (ISWC), (pp 130–145).

[22] D. Aum¨uller, H. H. Do, S. Massmann, and E. Rahm (205). Schema and ontology matching with COMA++. In Proceedings of the International Conference on Management of Data (SIGMOD), Software Demonstration.