

CAPITAL UNIVERSITY OF SCIENCE AND
TECHNOLOGY, ISLAMABAD



Schema Extraction and Integration of List Data from Multiple Web Sources

by

Umra Naeem

A thesis submitted in partial fulfillment for the
degree of Master of Science

in the

Faculty of Computing

Department of Computer Science

2018

Copyright © 2018 by Umra Naeem

All rights reserved. No part of this thesis may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, by any information storage and retrieval system without the prior written permission of the author.

To My Parents, Husband & Daughters



CAPITAL UNIVERSITY OF SCIENCE & TECHNOLOGY
ISLAMABAD

CERTIFICATE OF APPROVAL

**Schema Extraction and Integration of List Data from
Multiple Web Sources**

by

Umra Naeem

MCS151016

THESIS EXAMINING COMMITTEE

S. No.	Examiner	Name	Organization
(a)	External Examiner	Dr. Mehdi Hassan	AU, Islamabad
(b)	Internal Examiner	Dr. M. Tanvir Afzal	CUST, Islamabad
(c)	Supervisor	Dr. Nayyer Masood	CUST, Islamabad

Dr. Nayyer Masood

Thesis Supervisor

January, 2018

Dr. Nayyer Masood

Head

Dept. of Computer Science

January, 2018

Dr. Muhammad Abdul Qadir

Dean

Faculty of Computing

January, 2018

Author's Declaration

I, **Umra Naeem** hereby state that my MS thesis titled “**Schema Extraction and Integration of List Data from Multiple Web Sources**” is my own work and has not been submitted previously by me for taking any degree from Capital University of Science and Technology, Islamabad or anywhere else in the country/abroad.

At any time if my statement is found to be incorrect even after my graduation, the University has the right to withdraw my MS Degree.

(Umra Naeem)

Registration No: MCS151016

Plagiarism Undertaking

I solemnly declare that research work presented in this thesis titled “*Schema Extraction and Integration of List Data from Multiple Web Sources*” is solely my research work with no significant contribution from any other person. Small contribution/help wherever taken has been dully acknowledged and that complete thesis has been written by me.

I understand the zero tolerance policy of the HEC and Capital University of Science and Technology towards plagiarism. Therefore, I as an author of the above titled thesis declare that no portion of my thesis has been plagiarized and any material used as reference is properly referred/cited.

I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of MS Degree, the University reserves the right to withdraw/revoke my MS degree and that HEC and the University have the right to publish my name on the HEC/University website on which names of students are placed who submitted plagiarized work.

(Umra Naeem)

Registration No: MCS151016

Acknowledgements

First and foremost, I am grateful to Allah Almighty for his shower of blessings upon me and giving me the strength, ability and opportunity to conduct and successfully complete this research.

I offer my deep and sincerest gratitude to my supervisor, Dr. Nayyer Masood, Head, Computer Science Department, for his invaluable guidance and continuous support throughout this research study. I am deeply inspired by his enthusiasm, honesty and motivation. It was a great honor to work under his guidance. Without his guidance and support, this thesis would not have been completed.

I would also like to thank my respected teacher Dr. Arshad Islam for his able suggestions which helped me a lot during my research.

I am extremely thankful to my parents for their love, prayers, and care. I am also very much thankful to my husband, my daughters, my in-laws, and my sisters for their love, understanding, prayers and continuing support to accomplish this research work. Without their help and support, I would not be able to complete my thesis.

I take pride in acknowledging the support of all my teachers, my family members and my friends during this research study.

Abstract

Extracting structured from web lists is challenging as compared to web table. Existing approaches perform schema extraction, data extracting and data integration from web tables. Few techniques exist that extract schema and data from web list, however, none of the technique is found which performs data integration of web lists from different sources belonging to the same domain such as in the domain of Computer Science faculty of different universities.

In this thesis, faculty data in list format of 110 universities have been collected from web and stored in text file. Algorithm 1 and Algorithm 2 has been applied on text files containing source code of faculty list of each university. Algorithm 1 extracts all text written inside HTML tags and convert each HTML tag into temporary character “\t”. Algorithm 2 splits data elements on the basis of “\t”. After getting data elements of each faculty member, string “Next Record” is added. After applying algorithm 1 and algorithm 2, data and schema has been extracted. Extracted data contains both schema and data. In next step, algorism 3 is applied which performs schema matching. These matching separates schema and data. Instance matching algorithm in next step has been performed on data and data is classified into its corresponding attribute in the integrated table.

Results of proposed algorithms have been evaluated in three ways i.e. using quantitative analysis, query based validation approach and comparison with existing techniques. Algorithm 1 and algorithm 2 have been evaluated on random sample of 20% websites from dataset of 110 websites. Precision, Recall and F-measure of algorithm 1 and 2 is 100%. For evaluation of schema matching algorithm, 10 more websites have been collected from web and on sample of these 10 websites precision, recall and F-measure of this algorithm is 80%, 62%, and 69% respectively. On dataset of 110 websites, instance matching algorithm has been evaluated. Precision, Recall and F-measure of instance matching algorithm is 95%.

In query based validation approach, different SQL queries have been performed on integrated data and results are retrieved. Comparison with existing approaches

show that data integration of web list is not performed by existing approaches. The proposed technique performs all three steps i.e. schema extraction, data extraction and data integration of web lists.

Contents

Author’s Declaration	iv
Plagiarism Undertaking	v
Acknowledgements	vi
Abstract	vii
List of Figures	xi
List of Tables	xiii
1 Introduction	1
1.1 Web Lists	2
1.2 Text Separators	5
1.3 Key challenges of Extracting Data From Web Lists	5
1.4 Schema Extraction	6
1.5 Schema Matching	7
1.5.1 Schema Based Matcher	8
1.5.2 Instance Based Matcher	8
1.5.3 Linguistic Matchers	8
1.5.3.1 Name Matching	8
1.6 Schema and Data Integration	9
1.6.1 Data Value Conflicts	9
1.6.2 Schema Conflicts	10
1.6.3 Data Model Conflicts	10
1.7 Problem Statement	11
1.8 Research Questions	11
1.9 Purpose	11
1.10 Scope	12
1.11 Organization of the Thesis	12
1.12 Definitions, Acronyms, and Abbreviations	12
2 Literature Review	13
2.1 Comparison of Existing Techniques	22

3	Methodology	25
3.1	Data Collection	27
3.2	Input File Creation	27
3.3	Global Database	28
3.3.1	Faculty Table	29
3.3.2	University Table	29
3.3.3	Attributes Table	30
3.3.4	Attribute_Synonym Table	30
3.3.5	Taxonomy	30
3.4	Proposed Methodology	32
3.4.1	Research Question 1	32
3.4.2	Research Question 2	34
3.4.3	Research Question 3	37
3.4.3.1	Data Matching/Data Classification	37
3.4.3.2	Name	38
3.4.3.3	Attributes Classified Based on Taxonomy	39
3.4.3.4	Attributes Classified Based on Structure	42
3.4.3.5	Heterogeneity Issues	44
3.4.3.6	Semi-Automated Approach	45
3.4.3.7	Unmatched/Unclassified Text String	46
3.4.3.8	Faculty Search-A Web Application	48
4	Results and Evaluation	52
4.1	Distribution of Attributes	53
4.2	Quantitative Analysis	54
4.3	Results of Research Questions	54
4.3.1	Research Question 1	54
4.3.2	Research Question 2	56
4.3.3	Research Question 3	59
4.4	Query Based Validation	61
4.4.1	Query 1	61
4.4.2	Query 2	62
4.4.3	Query 3	63
4.4.4	Query 4	64
4.5	Comparison With Existing Approaches	65
5	Conclusion and Future Work	68
5.1	Conclusion	68
5.2	Future Work	69
	Bibliography	70

List of Figures

1.1	A web table with simple structure.	2
1.2	A web table with complex structure.	2
1.3	A web list showing view of Faculty data of National University (NU).	3
1.4	HTML source code of one of the records of NU.	4
1.5	A fragment of web list (with headings) of Bacha Khan University.	4
1.6	A Snapshot of Faculty member of Shaheed Zulfikar Ali Bhutto Institute of Science and Technology (SZABIST).	4
1.7	A fragment of faculty list of WARWICK.	5
1.8	A fragment of faculty list of Institute of Mathematics & Computer Science, University of Sindh, Jamshoro.	5
3.1	Architecture Diagram of Proposed System.	26
3.2	Extracting HTML code using Advanced Web Scraper.	28
3.3	Input file containing HTML source code.	28
3.4	Data of Attributes Table.	30
3.5	A fragment of “Attribute_Synonym” table data.	31
3.6	Algorithm to extract schema and data from web list.	33
3.7	Algorithm to clean data.	34
3.8	(a): Multiple values for Email, (b): Multiple values for Qualification.	36
3.9	(a): Missing values, (b): Missing values.	36
3.10	Schema Matching Algorithm.	37
3.11	Instance Based Matching Algorithm.	38
3.12	Research areas extracted from work of Hoonlor (Hoonlor et al., 2012).	41
3.13	Taxonomy of Research Interest in Taxonomy table.	41
3.14	A fragment of faculty record with same specialization and research interest.	42
3.15	Input Form for Administrator.	46
3.16	Input Form for Administrator.	47
3.17	Garbage Value.	47
3.18	Interface of Webpage of Search Faculty Information.	48
3.19	A Fragment of Records Searched by Designation (lecturer).	49
3.20	A Fragment of Records Searched by Specialization/Research Area as Artificial Intelligence.	49
3.21	fragment of records searched by University.	50
3.22	A fragment of extracted records by Name.	50

4.1	A Number of Attributes Instances on 110 websites.	53
4.2	Output Fragment of Applying Algorithm1 on GU List.	55
4.3	Output Fragment of Applying Algorithm2 on Output Of Algorithm 1.	55
4.4	Quantitative Analysis of Attributes.	61
4.5	Query 1.	62
4.6	Output of Query 1.	62
4.7	Query 2.	62
4.8	Output of Query 2.	63
4.9	Query 3.	63
4.10	Output of Query 3.	64
4.11	Query 4.	64
4.12	Output of Query 4.	64

List of Tables

2.1	A comparison of state of the art approaches.	22
3.1	A list of tables and its attributes in Global Database.	29
3.2	Taxonomy of Qualification, Designation and Research Interests. . .	31
3.3	Heterogeneity Issues in the domain of Universities' Faculty.	44
4.1	Precision, Recall and F-Measure of Instance Matching Algorithm. . .	57
4.2	Precision, Recall and F-Measure of Schema Matching Algorithm. . .	58
4.3	Precision, Recall and F-Measure of Instance Matching Algorithm. . .	60
4.4	Comparison with Existing Approaches	65

Chapter 1

Introduction

World Wide Web (WWW) contains huge amount of information in the form of unstructured, semi-structured and structured data. Structured data on web pages is usually presented in form of HTML tables (Cafarella et.al, 2008). In table, data is represented in two dimensional grids, in which column represents data fields (headings) and rows represent records. Figure 1.1 shows format of a table. Web tables now-a-days have become a common and popular model to represent structured data on web in a sense they resemble database relations. It has been used in almost every field such as Academics, Government, enterprise, weather forecasting, hospitals etc. for their data representation.

Hypertext Markup Language (HTML) is a publishing language which is used in WWW to publish information for global distribution. It is tag based language and declaration of each element type generally comprises of a start tag, content, and an end tag (Raggett et.al, 1999). Tables on web pages representing relational data are constructed using `<table>` tag (Cafarella et.al, 2008). A web table may have simple or complex structure (Liu et.al, 2003). A web table has simple structure if it has $m \times n$ grid of m rows and n columns where first row generally contains columns headings and rows below it contain data values (Figure 1.1).

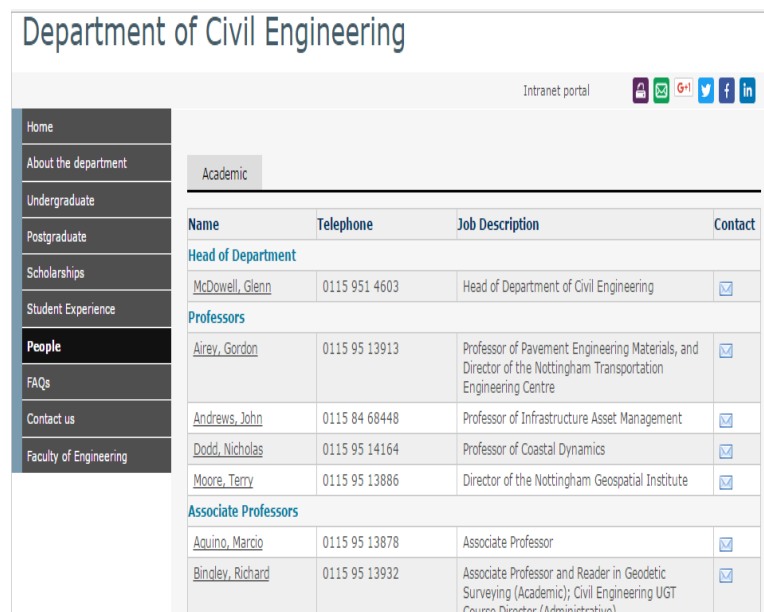
A complex web table may contain nested tables, rows containing single value as group header row. Figure 1.2 shows a complex table with group row headings.



The screenshot shows the website for the Department of Electrical Engineering, Islamabad Campus. The main content area is titled "Faculty Profiles" and contains a table with 10 rows. The table has three columns: "Sr. No", "Name", and "Designation".

Sr. No	Name	Designation
1	Dr. Muhammad Najam ul Islam	Dean (Engineering Sciences)
2	Dr. Saleem Aslam	Head of Department
3	Dr. Muhammad Ali Shami	Sr. Assistant Professor (On Study Leave)
4	Dr. Aft Raza Jafri	Sr. Associate Professor
5	Mr. Jehanzeb Ahmad	Associate Professor
6	Dr. Imtiaz Alam	Sr. Assistant Professor
7	Dr. Asim Ali Shah	Sr. Assistant Professor
8	Dr. Asad Waqar	Sr. Assistant Professor
9	Dr. Junaid Imtiaz	Sr. Assistant Professor
10	Dr. Aamir Shahzad	Assistant Professor (On Study Leave)

FIGURE 1.1: A web table with simple structure.



The screenshot shows the website for the Department of Civil Engineering. The main content area is titled "Academic" and contains a table with 10 rows. The table has four columns: "Name", "Telephone", "Job Description", and "Contact".

Name	Telephone	Job Description	Contact
Head of Department			
McDowell, Glenn	0115 951 4603	Head of Department of Civil Engineering	<input checked="" type="checkbox"/>
Professors			
Airev, Gordon	0115 95 13913	Professor of Pavement Engineering Materials, and Director of the Nottingham Transportation Engineering Centre	<input checked="" type="checkbox"/>
Andrews, John	0115 84 68448	Professor of Infrastructure Asset Management	<input checked="" type="checkbox"/>
Dodd, Nicholas	0115 95 14164	Professor of Coastal Dynamics	<input checked="" type="checkbox"/>
Moore, Terry	0115 95 13886	Director of the Nottingham Geospatial Institute	<input checked="" type="checkbox"/>
Associate Professors			
Aquino, Marcio	0115 95 13878	Associate Professor	<input checked="" type="checkbox"/>
Bingley, Richard	0115 95 13932	Associate Professor and Reader in Geodetic Surveying (Academic); Civil Engineering UGT Course Director (Administrative)	<input checked="" type="checkbox"/>

FIGURE 1.2: A web table with complex structure.

1.1 Web Lists

In addition to HTML tables, there exists a huge number of web pages that use lists to present structured data (Elmeleegy et.al, 2009). Lists can be generally used to represent ordered information, unordered information and definitions (Raggett

et.al, 1999). A list contains a series of similar type of data items or data records (Gatterbauer et.al, 2007). In this research, dataset consisting of web lists in the domain of Computer Science Faculty of different universities has been used. Figure 1.3 shows an example of web list in the chosen domain.

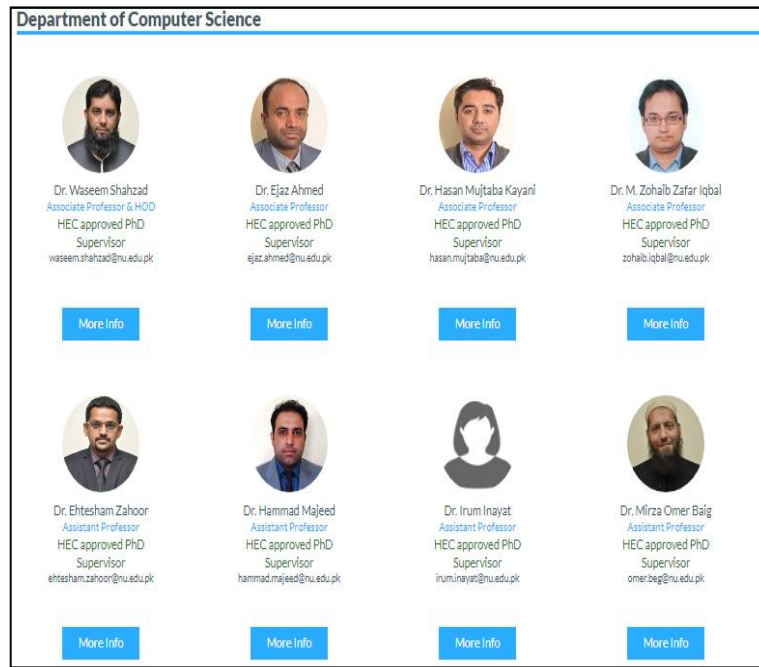


FIGURE 1.3: A web list showing view of Faculty data of National University (NU).

The HTML source code which is used to construct web lists is not only limited to the use of specific list tag such as ``, `` and ``. Now-a-days some other HTML tags such as `div`, `span` tag has also been used for providing layout of Web list. The HTML `div` tag provides a generic mechanism to add structure to documents and provides block level grouping of elements (Raggett et.al, 1999). Figure 1.4 shows HTML source code of one of the records of web list shown in Figure 1.3.

In this research, we have considered those web pages which represent faculty information as data records using different HTML tags such as ``, `<Div>`, ``, etc. in list format. In the domain of Faculty data, the web lists can be categorized into three types i.e. Web lists with Headings, Web lists without Headings, and Web lists with mixed representation. Web Lists with Headings includes headings along with all of its data records. Figure 1.5 is an example of this type of list.

```

<div class="staff-item">
  <div class="staff-item-wrapper">
    <div class="staff-info">
      <div class="staff-avatar">
        
      </div>
      <div class="staff-name">Dr. Waseem Shahzad</div>
      <div class="staff-job">Associate Professor & HOD</div>
      <p class="text text-success no-margin-bottom">HEC approved PhD Supervisor</p>
      <div class="staff-email">waseem.shahzad@nu.edu.pk</div>
    </div>
  </div>
</div>

```

FIGURE 1.4: HTML source code of one of the records of NU.

Name	Mr.Dilawar Shah
Designation	HOD, Assistant Professor
Email	dilawar_shah@yahoo.com
Phone no	0916540063
Qualification	MS

FIGURE 1.5: A fragment of web list (with headings) of Bacha Khan University.

In Web Lists without Headings, all data records are embedded on web pages without any heading. Figure 1.6 shows web list without heading.

Dr. Mohammad Altaf Mukati
 Vice President (Academics) & Dean (Computing & Engineering Sciences)
 Ph.D (Hamdard University)
 Engineering Sciences

FIGURE 1.6: A Snapshot of Faculty member of Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology (SZABIST).

Figure 1.7 shows a web list with mixed representation. In each data record, some data entries are with heading and some without headings.



FIGURE 1.7: A fragment of faculty list of WARWICK.

1.2 Text Separators

Text separators are used as a delimiter between data values within each data record. Text separator can be a sequence of sequential HTML tags in case each data value is appearing on single line. See Figure 1.4 for such records.

In some records, text separator within a line can be a punctuation mark like ‘,’, ‘:’, ‘()’. Figure 1.8 shows an example of such data record.

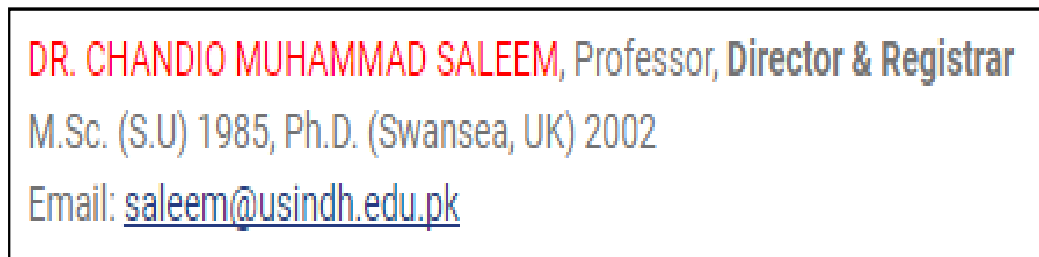


FIGURE 1.8: A fragment of faculty list of Institute of Mathematics & Computer Science, University of Sindh, Jamshoro.

In Figure 1.8, designation is written with attribute Name, so we can split it on the basis of specified punctuation mark.

1.3 Key challenges of Extracting Data From Web Lists

Most of the existing techniques (Purnamasari, et al., 2015; Gatterbauer et.al., 2007; Embley et al., 2005; Gultom et al., 2011) extract schemas of web tables in

which column headings are usually represented in first row while remaining rows act as data rows. Hence, tables' schema is extracted from first row easily. However, schema extraction from lists as compared to tables is a challenging task because of the following reasons:

1. Tables have a fixed number of rows and columns. (e.g. See <http://bamu.ac.in/dept/csit/faculty.htm>) Whereas, HTML lists may have variable number of rows and columns. (e.g. See <http://www.abasynisb.edu.pk/facultycs/index>)
2. In tables, headings are specified only once, whereas in lists containing column headings, headings are usually repeated for every data occurrence. (e.g. See <http://www.bkuc.edu.pk/welcome/department/Computer%20Science/20>)
3. Mostly, web lists are used with mixed representation.
4. Some lists may only have data instances. Headings are not specified for them.
5. For tables, only `<table>` tag is used, but for lists, a varying number of tags can be used.
6. Data values of multiple attributes may be specified in a single HTML tag.
7. Different web sites may use different format for list representation.

The approach proposed in this research works on list data coping with all these challenges.

1.4 Schema Extraction

In all databases, structure and the formal semantics of the possible instances are defined by a term called schema (Biskup, 1995), which is needed to manage the information stored in the database. In a relational database, schema is simply a collection of tables and generally a separate or a single schema is created for

each table (Blinn et al., 1999). In this context, a table schema defines the table name, attribute names, domains i.e. data types for columns (Trker, 2001). But, this information is not explicitly provided as table metadata with web tables; human users can easily understand it by following information specified in labeled columns (Cafarella et al., 2008). In web tables and web lists context, schema is only contained in attributes' name, which may also be missing in many cases. Techniques and tools are needed to extract these attribute names from web tables and web lists.

Schema extraction in web is task of extracting schema i.e. attributes of the tables from web tables. Data in web tables is generally loaded from the databases stored at host's end (Zhai, & Liu, 2005). Tables' schema is not explicitly provided with the HTML tables. So in order to perform different operations such as search queries, business competitive analysis etc., there is a need to extract tables' schema and data from web pages and store them at a centralized location i.e. in a single database to perform such types of operations. Three steps i.e. schema matching, data extraction and data integration are required to integrate data from multiple web sources in a single location. Following is the detailed description of these steps.

1.5 Schema Matching

Schema matching is a process of mapping attributes of two schemas in which semantic match exists (Wang et al., 2004). It has become a fundamental problem in various database application domains; which include data integration, data warehousing, e-business, and semantic query processing (Rahm et al., 2001; Berlin & Motro, 2002). As schemas are created independently by different people with different real world scenarios, so they differ widely in terms of structure and terminology (Rahm et al., 2001).

1.5.1 Schema Based Matcher

Schema based matcher can only use schema information such as attribute names for the purpose of matching schema. In schema based matcher, element-level and structure-level matching can be performed. Element-level can further utilize Linguistic or Constraint-based matching, whereas, Structure-level can only use Constraint-based matching. (Rahm et al., 2001).

1.5.2 Instance Based Matcher

In instance based matcher, only element-level matching can be performed which further can use Linguistic or constraint-based matching techniques. Instance-level data, utilizes the contents and meaning of schema elements and can help to construct schema either manually or automatically for the cases in which either scheme information is limited or not present at all (Rahm et al., 2001).

1.5.3 Linguistic Matchers

Linguistic matchers also called language-based matchers use name and other textual information to check the similarity between schema elements. (Madhavan et al., 2001; Rahm et al., 2001).

1.5.3.1 Name Matching

Some of the techniques that can be used for Name Matching are given below.

- Names Equality
- Canonical Name Equality
- Synonyms Equality
- Hypernyms Equality

- Similarity of names based on common substrings, edit Distance

In this research, we are using both schema based and instance based matchers. Instance based matching is required because we have found very few websites which are fully representing their faculty data with attributes names i.e. schema. Many websites have missing schema for some of the attributes, there also exists many websites which do not schema at all. So, in order to cope with such varying formats, schema based matchers are not found to be enough so instance based matching is also required for correct identification of data values.

1.6 Schema and Data Integration

Schema Integration is the process of combining schemas of multiple databases either existing or proposed into a single, combined, global schema (Batini et al., 1986; Elmagarmid et al., 1999). So, after completing the process of schema matching, matching elements are combined under a global schema. After schema matching, if any schema element does not find any semantic match with other schema, simply put this element in integrated schema (Devogele et.al, 1998).

Data Integration is the process of integrating data from multiple web sources at a single location. Web data integration is a hard task due to dynamic and heterogeneous nature of web data. Heterogeneity conflicts arising from use of multiple web data sources can be classified into three main categories. (Hajmoosaei et.al, 2008; Rahm and Do, 2000).

1.6.1 Data Value Conflicts

They arise at instance level and are associated with the representation of the data values.

1.6.2 Schema Conflicts

They occur due to the use of different schema used by different web sources modeling the same real world scenario.

1.6.3 Data Model Conflicts

Data modeling conflicts are the conflicts which arise as a result of using different data models e.g. relational model for one database and object-oriented model for another database.

In this research, we are only handling two heterogeneity conflicts i.e. schema conflicts and data conflicts. In case of schema matching of web lists, there are chances of finding more heterogeneity issues than web tables due to the following reasons.

1. We do not have proper defined schema.
2. Data is semi-structured.
3. Inter-mixing of schema and data.

As described earlier, data of similar domain is represented using different formats on different web sites. For example in Figure 1.1, 1.2, 1.3 and 1.5, faculty list of different universities has been represented in different formats, although they belong to similar domain. Due to variations in data formats, and use of different tags for representing structured data, having a lot of heterogeneity issues, extracting schema and data from web lists has become a challenging task and this area has gained an increasing attention over the last few years.

1.7 Problem Statement

Most of the existing schema extraction approaches extract only web tables constructed using `<table>` tag. Relatively less work has been done for schema extraction from web lists and none for the data integration. The proposed technique is focused on schema extraction and data integration from web lists constructed with different html tags such ``, `<div>`, ``, etc. The goal of this research is to extract schema of web lists, perform schema matching, extract data and integrate the extracted data from multiple web pages into a single table.

1.8 Research Questions

Following research questions will be addressed in this research.

1. How accurately can we extract schema/data from web lists dealing with their inherent variations?
2. To what extent can schema matching be performed on schema extracted from web lists?
3. Can extracted data from web lists be integrated into database table?

1.9 Purpose

The purpose of this thesis is to extract such web pages from faculty domain which have represented faculty data in list format. After collecting web pages, schema and data of web lists from each web page is extracted, and integrated into single relational table so that query processing can be performed on the integrated data.

1.10 Scope

The proposed solution will be highly beneficial for research community as data of a single domain will be available at centralized location and user can perform search queries on this centralized data to extract their required information.

1.11 Organization of the Thesis

This document is comprised of five chapters. Chapter 1 contains introduction to the problem, research questions, and purpose of the study. Chapter 2 contains comprehensive literature review of state-of-the-art approaches related to the problem described in chapter 1. In Chapter 3, detailed methodology to the solution of the problem has been stated. Chapter 4 includes results and their evaluation. Chapter 5 provides conclusion and future work.

1.12 Definitions, Acronyms, and Abbreviations

Term	Definition
Table Schema	It states the table name, name of each column, and the data types of these columns etc (Türker, 2001).
Schema Matching	It is a process of mapping schemas of two tables between which semantic match exists (Rahm et al., 2001).
Data Integration	Combining data from multiple sources at a single location (Lenzerini, 2002).

Chapter 2

Literature Review

Processing web tables and web lists has become an active area of research now-a-days. Numerous efforts have been made in order to extract web tables' schema and data from multiple web sources. Existing approaches are classified into manual, supervised, and unsupervised techniques. These approaches are based on HTML source code, visual information of contents used in web pages, and Document Object Model (DOM) trees etc. Most of the research touches the area of schema extraction and data extraction from web tables but very less effort has been done in the area of data integration. Following is the review of some state-of-the art techniques.

In 2015, a technique comprising of three algorithms was proposed in order to extract HTML tables from Web (Purnamasari et al., 2015). These algorithms are run in sequential manner i.e. first algorithm 1 is executed, then algorithm 2 and in last step algorithm 3. The purpose of Algorithm 1 is to find number of rows and columns of the table which is helpful in determining the table size. The total number of rows are calculated by adding all `<tr>` tags present inside the `<table>...</table>` tag. The total number of columns is calculated by adding all `td` tags found in first `<tr>` tag. The number of colspan will be counted if `<tr>` tag contains colspan.

The second algorithm finds table property i.e. how many rows occupy column headings. This is done by determining maximum value of rowspan in each of the `<td>...</td>` tag in tag `<tr>...</tr>`. Once table size and table property has been extracted, Algorithm 3 is applied to extract the table contents. The algorithm 3 works by starting from row border (property) found in algorithm 2 and goes until it reaches first row. Here, at each iteration, row value is decremented. Column heading are then extracted using val of colspan. The authors have tested these algorithms on a sample set of 100 HTML tables. Results obtained from these experiments have been evaluated using precision, recall and F-measure. This technique has been tested on HTML tables. It has not been tested for real websites table. This is the simplest and recent technique of schema extraction of web data however it has some limitations in that it only extracts table schemas and identifies the table area from where data region is starting, but it does not actually extract data. Another observation about above mentioned technique is it is only applicable to simple tables. Tables with complex structure such group header rows after start of data rows are not handled with this method. Moreover, it does not consider linked page information present in simple structure.

An unsupervised learning approach which performs page level data extraction is proposed by Krishna and Dattatraya (Krishna, & Dattatraya , 2015). A website may present its information in two ways. One is to use fixed size templates, while other is to use variant size templates for all pages of the same website. This approach extract schema and data from template generated web sites using visual information present on web pages for example, background color, text position, border style etc. The proposed system operates in different steps. In first step, input as two web pages is provided to the system. A vision-based page segmentation algorithm is then applied on each page which segments web page into different segments and builds a Visual Block (VB)/Document Object Model (DOM) tree. Blocks in VB tree are then compared to find fixed or variant template pages. A noise-block-removal algorithm is then applied to remove noise blocks from DOM trees. After this step, DOM trees for fixed template pages are merged using tree

merging algorithm. Variant tree matching algorithm is used to merge variant template pages. Pattern tree is built from these merged trees and scheme is extracted from pattern tree. Data is then extracted by comparing and matching pattern tree and HTML tree. The authors have tested their system on ten websites and performed time based comparison of their system with existing system. The existing system includes noisy blocks in DOM tree whereas proposed system works by removing those noisy blocks and works by taking less time hence making system efficient. However, there is need to test the proposed system on large dataset to test its accuracy and efficiency.

Adelfio and Samet (Adelfio & Samet, 2013) propose a new approach of schema extraction which involves classification technique based on supervised learning. It uses concept of “Conditional Random Fields (CRF)” which is used for the task of sequence labeling. First they used the technique used by Cafarella (Cafarella et al., 2008) to classify a table as relation or non-relational. This technique first defines a set of row classes. Individual letters are used to represent each row class. Letter ‘H’, ‘D’, and ‘T’ are used to represent header row, data row and title row respectively. Group header row is represented by letter ‘G’, similarly row containing aggregate values such as “Total” are labeled with letter ‘A’. To denote Non-relational metadata rows, and blank rows letters ‘N’ and ‘B’ are assigned respectively. The authors reported the use of logarithmic binning scheme for the encoding of set of individual cell as attributes row features. Cell attributes are divided into three categories; layout attributes, style attributes, and value attributes. The proposed algorithm takes a table as input, extracts its cell attributes, then it computes row features, and finally rows are classified into known classes. The supervised classification method in combination with CRF determines the correlation between row labels and row features. CRF then uses this information for assigning labels to testing data set. The CRF returns sequence of row labels as output and from this output, schema for relational table is extracted. Dataset used for experiments includes both HTML tables and spreadsheets table. The authors show that their technique provides better results than a popular method (Cafarella et al., 2008) which is used to extract schema. The problem with this technique is

large training data set is created using hand annotated data tables which is time consuming and labor intensive. Query processing on noisy data extracted from web tables is the area of their future work. They have also mentioned future work of conversion of web tables into a fully relational (normalized) structure.

Data integration of web data is performed by Gultom, et al., (Gultom et. al, 2011) by developing Mashup, a web application which is used to integrate data from various web pages. Mashup performs this task in different phases. First phase includes extracting data from different web sources. In second phase, data modeling is performed, and then in next stage, on this data, cleaning process is applied. After this step, data integration is performed and then last step of data visualization is carried out. The authors introduce a new application system “Xtractorz”.

The Xtractorz application has been tested on two tables available on web page of result of National General Election, Indonesia of 2009. In first step of data extraction, the Xtractorz system extracts data from web tables and present HTML tags into DOM tree using recursive algorithm. In Second step of data modeling, a structured form of DOM tree has been formed, this step is performed automatically i.e. the system considers two tables sharing at least one column and placing different columns in single table. The next stage of data cleaning also called data filtering filters the extracted data and makes corrections if required. The issues such as spelling mistakes, problems in content alignment and format conversion issues are resolved in this stage. After this step, data from different web tables is integrated and stored into a single table. The proposed system has been compared with RoboMaker and Karma systems and results shown that proposed system is efficient then these two systems.

Nagy et.al, (Nagy et.al, 2011) propose a relatively new approach for factorization of web tables which is based on indexing technique. The indexing of data cells is provided through Row and Column header hierarchy. The concept of relational algebra has been used to represent collection of row and column header paths as sum-of-products expression. Web tables are extracted from large statistical

websites and their CSV version has been created, Header paths have then been extracted from CSV file. After extracting table headers, and factorizing it into canonical representation along with data cells, table can easily be converted into relational table for relational database. Different SQL queries have been performed on this relational table. Out of 1000 web tables, experiments have been performed on 107 randomly selected web tables. In September 2011, Nagy et.al, (Nagy et. al, 2011) extended this work by presenting an approach to deal with more complex table layout. Some new experiments are performed and an interactive tool VeriClick is also introduced for table correction. The dataset contains 200 web tables from ten large statistical websites. Using this approach of indexing through header paths hierarchy, 376 relational tables are generated and 34,110 subject-predicate-object RDF triples. In their work, data integration is not performed.

Elmeleegy et.al, (Elmeleegy et.al, 2009) proposed a technique LISTEXTRACT which finds best possible relational table that can be created from list. This technique works in three steps which are independent splitting phase, alignment phase and refinement phase. In independent splitting phase, each line is converted into records with multiple fields. The concept of Field Quality Score (FQ) has been used in order to assess the quality of a particular field as cell value. The FQ has been calculated using Type Score, Language Model Score and Table Corpus Score. In Alignment phase, an initial table T1 is created and number of columns for this table is determined by considering most common number of fields in all records. Both long and short records are aligned in this step. Long records with more than 'k' fields are re-split in such a way that they have no more than 'k' fields in their new updated records. Short records are aligned using insertion of NULL values. In Refinement phase, fields assigned to T1 are analyzed to find and correct incorrect field assignments.

Two datasets have been used for experimentation purpose. One dataset contains 20 HTML lists from varied domains available on Web. The other dataset includes 100 lists formed from 100 HTML tables. The authors have compared their technique with Road-Runner (Crescenzi, Mecca & Merialdo , 2001) which is heavily used by researchers. The authors favored their technique over Roadrunner using

precision, recall and F-measure. The limitation of this technique is that it only extracts data, converts it into number of rows and columns i.e. relational table but it does not extract schema and assign column headings to extracted data. Further, it does not perform data integration.

Gatterbauer et.al. (Gatterbauer et.al, 2007) devise an information extraction system for web tables that is domain independent. The approach is based on the two-dimensional visual box model which web browsers use to display the information on the screen. Authors made an observation that most of the web tables topologically create a frame in the visual box model. The visual and style information gained using this approach eliminates the gap generated due to missing domain-specific knowledge about table templates and its contents. The approach has been named VENTex for Visualized Element Nodes Table extraction. Given a web page, VENTex first analyzes it to detect table, using their spatial arrangement, relations are recognized. In next step, rows of table are extracted along with hierarchical information of relations between their entities and then it is saved in XML format. For testing of proposed extraction system, authors created table ground truth from a wide variety of web tables. After this step, 493 web tables from 269 web pages were created for ground truthing. Domain independency has been achieved by providing a test set of web tables from different domains collected by 63 students. The recall and precision of table extraction was 81% and 68%, respectively. The recall and precision of table interpretation was 57% and 48% respectively. In their work, schema matching and data integration has not been performed.

The problem with HTML tables is that direct queries cannot be performed on them due to their unknown structure. This problem is studied by Embley et al. (Embley et al., 2005). In this work, they have proposed a solution for this problem relying on document-independent extraction ontology. The components of an extraction ontology are an object/relationship-model instance, and a data frame. The object/relationship-model instance defines sets of objects, their relationships, and constraints on the sets of object and relationship. The possible contents of the object set are described in data frame. The proposed approach consists of

three steps: table understanding, data integration, and wrapper generation. Tables of interest can be found from Web pages using step of Table understanding. This step further includes tasks of attributes and values identification, and then forming data records by pairing attributes and its values. The linked pages within tables are also considered. The step of Data integration matches source records with a target schema. Using wrappers, data is extracted from source records and is stored in a target schema. The experiments have been performed on datasets of car advertisements and cell-phone sales. The results show that data of interest from tables in above mentioned domain has successfully been extracted and has been transferred from source HTML tables to a given target database table on which direct queries can be performed.

Zhai and Liu (Zhai, & Liu, 2005) propose a technique to extract data records from web pages. This work is based on two steps. In first step, the authors use their previous technique Mining Data Records (MDR) (Zhai et al, 2003) with an improvement to identify data regions and data records. The improved technique is given the name MDR-2. MDR technique is based on two observations: First observation is similar type of data records are presented in some specific area of web page and follow same formatting using similar type of HTML tags. The second observation is that in tag tree data entries presented in some specific region using similar tags are shown under one parent node. The proposed technique works by performing three steps. In first step, it builds tag tree of the given web page whereas in second step, it uses tag tree and string matching algorithm to identify data regions of page and after identifying data regions, it identifies data entries from those regions in third step. The proposed technique finds both contiguous and non-contiguous data records contained within a web page. In order to extract data regions, tag strings of individual nodes and combinations of multiple adjacent nodes are compared. Each similar node (tag) and each node (tag) combination is represented as generalized node. The contiguous generalized nodes represent a data region. This technique only identifies data region and data records, it does not extract data records. As their future work, they propose to find data records that are not formed by HTML table related tags.

MDR-2 works the same as MDR but improved in a way that it also uses visual information to identify data region. The part of visual information focuses on the observation that distance between two regions should be larger than the distance within any data record. In second step, partial tree alignment algorithm, which uses concept of tree matching, is used for data extraction. It only matches tags of tree, data is not compared. This process is also performed using two steps. In first step, one rooted tag tree against each data record is created. After finding such sub trees for all data records, all sub trees are combined into a single tree. In second step, partial tree alignment algorithm is used to align tag trees of all data records contained in each region. The concept of seed tree to align these multiple tag trees is introduced and this seed tree grows increasingly. A tree with maximum number of data fields is selected as seed tree. Partial tree alignment of two trees T_i and T_s is done node by node. The nodes are compared in two trees T_i and T_s , if they match then a link is created between them and they get aligned, and are inserted into data table as single column. If they do not match then this node is added in the seed tree. After completing the whole process for all tag trees, if some data is unmatched, then for each unmatched data, a separate column will be created.

A system called, DeLa, which sends queries through HTML forms and gets set of retrieved web pages as a result of queries has been introduced by Wang and Lochovsky (Wang & Lochovsky, 2003). It generates regular expression wrappers to extract data from retrieved pages and stores extracted data into a table. Meaningful Labels i.e., the columns of the table are then assigned to extracted data. The authors only considered those web sites for which web pages are generated dynamically i.e. by querying the data stored in a back-end database. The four modules of DeLa system are: a form crawler, a wrapper generator, a data aligner and a label assigner. For form crawler, existing hidden web crawler, HiWe by Raghavan and Garcia (Raghavan & Garcia, 2001) has been used. In wrapper generator part, regular expression wrappers are automatically generated from data present in web pages. In data aligner module, data from regular expression wrappers is extracted and stored in a table. After this attributes are separated. For label assignments, four

heuristics are used which are: match form element labels to data attributes, search for voluntary labels in table headers, search for voluntary labels encoded together with data attributes, label data attributes in conventional formats. Results show the system provides more than 90% correctness for data extraction and about 80% correctness for label assignment.

Lerman et al., (Lerman, et al., 2001) introduce an approach which extracts data from lists and tables and groups the extracted data by rows and columns. The technique works by splitting text of individual web pages into tokens and each token is assigned its syntactic type i.e. the token can be a punctuation, an alphanumeric or an HTML token. The approach works in three steps. In first step, data is extracted from lists, for this page template is computed and list is identified. After this a set of features are computed which include separators and content. In second step columns are identified; AutoClass tool has been used to classify data into columns. In third step, grammar induction of regular languages has been used for rows identification. Limitation of this approach is that it needs multiple pages to be analyzed from same source before data extraction. This approach does not work for single list given on single page.

Yoshida, et al., (Yoshida et al., 2001) highlighted the issue of using different communication styles on WWW for similar domain on different websites. The proposed system comprises of two steps which are Table Structure Recognition and Table Integration. They define table structure (table type) as layout of attributes and values. They defined nine different table types in this work. Given a table, Table Structure Recognition determines the part of table as attribute or value. Ontological knowledge about different objects in different formats have been extracted and then used in the process of table recognition. Expectation Maximization algorithm (Dempster, et al., 1977) has been used for this task.

After performing this step, table structures from many tables have been recovered. Then, step of table integration starts and it integrates tables of similar domains but with different formats into a single table. This task is performed using two steps, first it is decided which tables should be part of which integrated table.

This is done by making clusters of tables of same category. Second step of Table Merging is applied on each cluster to integrate all tables of the cluster into one. As it is possible that one attribute may be represented with some other title in other tables so attribute clustering method has been applied to handle this issue. At the end, each cluster contains large table with objects of similar category.

The Table Structure Recognition algorithm has been applied on 35232 tables and accuracy of this algorithm has been calculated from random sample of 175 tables. The result of precision and recall has been compared with technique proposed by Chen, et al., (Chen et al., 2000). The results of precision for proposed technique are not better than Chen, et al., (Chen et al., 2000) technique. According to authors precise comparison of their technique cannot be performed due to difference in nature of data used. Overall accuracy of complete approach is 78%.

2.1 Comparison of Existing Techniques

Following table shows the comparison of some existing approaches.

TABLE 2.1: A comparison of state of the art approaches.

S. No.	Authors/Year	Dataset Type	Schema	Data	Data	Technique Used
			Extraction	Extraction	Integration	
1	Purnamasari, et al., 2015	HTML Tables	Yes	No	No	Wrapper induction based
2	Krishna, & Dattatraya , 2015	Web tables and web lists	Yes	Yes	No	Unsupervised learning approach, based on DOM trees and visual cues
3	Adelfio & Samet, 2013	HTML tables and spreadsheets table	Yes	Yes	No	Classification technique based on supervised learning

S. No.	Authors/Year	Dataset Type	Schema Extraction	Data Extraction	Data Integration	Technique Used
4	Gultom et al., 2011	Web tables	Yes	Yes	Yes	DOM tree based
5	Nagy et.al., 2011	Web tables	Yes	Yes	No	Index based
6	Elmeleegy et al., 2009	Web lists	No	Yes	No	Unsupervised learning, language model and HTML table corpus
7	Gatterbauer et.al., 2007	Web tables	Yes	Yes	No	Visual box model
8	Zhai, & Liu, 2005	Web Tables	Yes	Yes	No	HTML tag tree based on Visual information
9	Embley et al., 2005	Web Tables	Yes	Yes	Yes	Ontology based
10	Wang & Lochoovsky, 2003	HTML forms	Yes	Yes	No	Regular expression wrappers
11	Lerman et al., 2001	Web tables and lists	Yes	Yes	No	Unsupervised learning algorithms

The comparison has been performed on the basis of some parameters which include Dataset type, Schema Extraction, Data Extraction, Data Integration, and Technique used. “Dataset Type” defines whether technique has been tested on HTML table, web table, spreadsheet table or web list, etc. The values “Yes” or “No” in column “Schema Extraction” shows whether this technique extract schema of web tables, spreadsheets, web lists etc. or not. Similarly, the values “Yes” or “No” in column “Data Extraction” shows whether this technique extracts data of web tables, spreadsheets, web lists etc. or not. And, the values “Yes” or “No” in column “Data Integration” shows whether this technique performs data integration or not. The parameter “Technique Used” shows the featured and technique used by proposed approach.

Table 2.1 shows that existing techniques extract tables' schema and data but less work has been done in the area of data integration. Most of the existing techniques work for web tables which are constructed using Html <table> tag. Some approaches exist which are developed for web lists constructed with varying tags such as <div>, , etc. but those approaches are generally able to extract schema or data of web lists. These techniques do not perform schema matching and data integration. The techniques which are used to extract lists schema cannot be applied straight forward on the domain of faculty data due to extreme heterogeneity issues and diversity in list formats.

Chapter 3

Methodology

This chapter provides detailed description of overall methodology that has been used to conduct this research. The aim of this research is to extract faculty data from web lists, extract their schema and finally store and integrate data from multiple web sites in a single database table. Using this integrated data view, users can specify queries to find data of their interest.

The approaches developed thus far focus on schema extraction, data extraction, and data integration of Web Tables. Similarly, few techniques are also available to extract schema and data from web lists but none of the technique has been found which performs data integration of web lists data. World Wide Web is heavily used by people for getting information about anything. Universities all over the world are also using internet to share data of their universities so that people in any part of world can get information about the university and even apply online. The information which universities provide over the Web are generally admission details, faculty information, contact information etc. Faculty is an important part of a university, when a student intends to take admission in any university, faculty profiles are visited to find a match between student's research interests and other details. So, many universities present their faculty data on web pages. Faculty data may be represented using web tables or web lists. In this thesis, web pages showing faculty data of different universities in list format has been taken for experimental data.

A faculty list of a university may be different from some other university list in terms of its visual format, use of different HTML tags in source code. Further, on single webpage, number of HTML tags may vary from record to record. Due to these differences, set of tags used for records representation are not consistent. To make records consistent i.e. one record per line, a Google Chrome extension has been used. After making records consistent, text is extracted from each line; a single line holds the record of one faculty member. For Schema matching, set of existing attributes and their synonyms has been used. Here, for web data, term “Schema” refers to attributes names as in case of web data, schema generally includes only column headings. For data matching, taxonomy for different attributes has been built. After mapping data values to associated attribute, data of each website is stored in a single global table thus performing the implicit integration.

Following is the architecture diagram of proposed methodology.

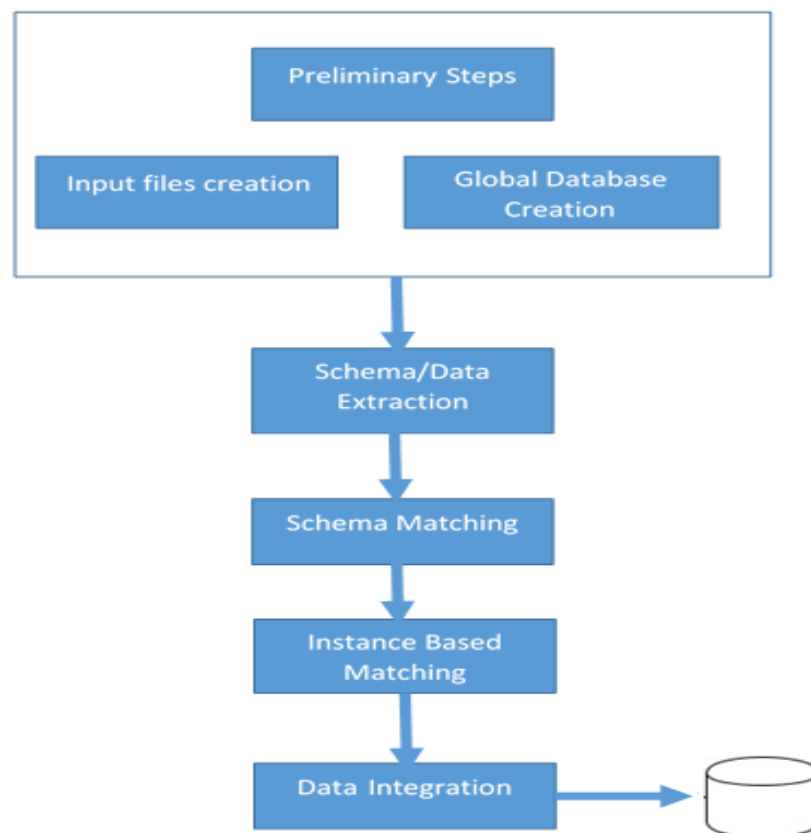


FIGURE 3.1: Architecture Diagram of Proposed System.

3.1 Data Collection

This research has been conducted on web lists in the domain of Faculty Data in the Department of Computer Science. The focused department is Computer Science, but some other departments are also considered due to intermixing of Computer Science with other disciplines such as Software Engineering (SE), Information Technology (IT), etc.

The dataset has been collected manually from different universities' websites. It covers universities from different countries of the world such as Pakistan, India, UK, Australia, China, and USA etc. In first step, websites of different universities have been extracted and particular web pages with their faculty members' list have been identified. Then, all these web pages have been analyzed to filter out such pages which are representing their faculty information in list format. If the faculty information is embedded in web page according to our required format, then it is added into our dataset otherwise it is ignored.

3.2 Input File Creation

For each web page, input file in .txt format for Algorithm 1 has been generated by extracting HTML source code of data regions of interest. In order to extract HTML source code from web page, a Google Chrome extension: Advanced Web Scrapper has been used. This is an advanced web scraping app provided by Google Chrome for screen scraping using CSS selectors. Following are some advantages of using this extension:

1. Due to ill-formatting of HTML source code in many web pages, it was a difficult and time consuming process to extract source code of particular area of web page from a large code HTML file. Google Chrome extension minimized this effort and, hence, our task of source code extraction became quick and easy.

- Another advantage of using this extension is that it returns source code of all web pages in consistent way. It organizes record of each faculty member in single row making it to be easily read and analyzed by the algorithm.



FIGURE 3.2: Extracting HTML code using Advanced Web Scraper.

Figure 3.3 shows some part of input file containing source code of faculty webpage of Abbottabad University of Science & Technology.

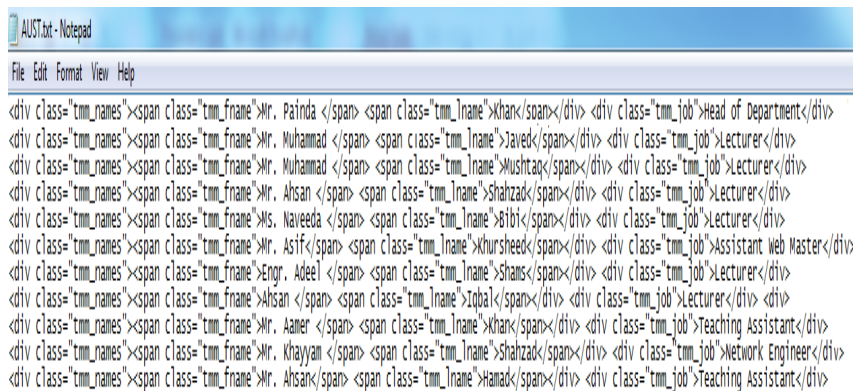


FIGURE 3.3: Input file containing HTML source code.

3.3 Global Database

A global database “Academic Staff List” has been created in SQL Server 2008. The following table shows tables of this global database along with their attributes.

TABLE 3.1: A list of tables and its attributes in Global Database.

Table	Attributes
Faculty	S. No., Name, Designation, Department, Uni_id, Campus_id, Qualification, Email, Phone No, Office_Extension, Fax, Specialization, Research_Interest, Room_No, Job_Status, Semester, Web_Profile
University	Uni_ID, Campus_ID , University_Name, Short_Name, Campus_City, Country, Address, Contact_No, Website_URL
Attributes	Attribute_Code, Attributes_Name
Attribute_Synonym	Attribute_Code, Attributes_Synonyms
Taxonomy	S. No., Designation_Taxonomy, Qualification_Taxonomy, Department_Taxonomy, Research_Interests

Following is the detail of tables given in Table 3.1.

3.3.1 Faculty Table

Faculty table is the target table in which data extracted from web lists will be integrated.

3.3.2 University Table

This table contains data of all the universities which have been used in dataset of the thesis. Name of each University, Campus City, Country, Address, Contact details and URL of university website has been manually collected from each website and then stored in University table.

3.3.3 Attributes Table

“Attributes” table contains attributes and their codes in the domain of Teaching Faculty. The following table contains list of initial set of attributes along with their codes. While applying the proposed algorithms on some new website, if some new attribute is identified, it is added to this table by the administrator.

Attribute_Code	Attributes_Name
NA	Name
DES	Designation
DEP	Department
QU	Qualification
EM	Email
PN	PhoneNo
EX	Office Extension
JS	Job Status
RI	Research interest
SP	Specialization
WP	Web Profile
RO	Room No
SE	Semester
FX	Fax

FIGURE 3.4: Data of Attributes Table.

3.3.4 Attribute_Synonym Table

“Attribute_Synonym” table consists of synonyms that can be possibly used for attributes of target table. A fragment of “Attribute_Synonym” data is given below.

3.3.5 Taxonomy

In this research we have used taxonomy based approach which stores vocabulary related to Qualification, Designation and Research Interest attributes in Taxonomy

Attribute_Code	Attributes_Synonyms
DEP	division
DEPT	department
DEPT	dept
DEPT	school
DES	designation
DES	position
EM	email
EX	ex
EX	ext
EX	office exten
JS	job status
JS	status
PN	cell
PN	office phone
PN	ph
PN	phone
PN	phone no
PN	pno
PN	t
PN	tel
PN	telephone
QU	qualification
QU	qualifications
RA	area of interest
RA	areas

FIGURE 3.5: A fragment of “Attribute.Synonym” table data.

table. Table 3.2 shows data of each attribute of Taxonomy table. For Research Interest attribute, 552 values have been stored, but due to space limitation, here some of them are shown. List of complete values of research interest attribute used in this research is available in the work of Hoonlor (Hoonlor, 2012).

TABLE 3.2: Taxonomy of Qualification, Designation and Research Interests.

Attributes	Taxonomy
Qualification	staff, chancellor, vice, professor, lecturer, assistant, associate, head, hod, chairman, chairperson, chair, director, dean, reader, programmer, instructor, tutor, supervisor, coordinator, manager, incharge, teaching fellow, teacher, visiting, principal ,junior, senior, engineer, administrator, researcher, lab, attendant, officer, ldc, advisor, co-ordinator, section, teaching, adjunct, specialist, member, fellow, scientist, emeritus, technologist ,technician, secretary, president

Designation	phd, MS, msc, mcs, BS, bsc, bcs, pgd, doct, masters, university, mphil, post doc, doctoral, mscs, bscs, BIT, MIT, ME, BE, diploma, MA, MED, bed, scholar, bachelor, master, bachelors, under graduate, post graduate, BBA, MBA, MEd, CCNA, MCSE, postdoc, MS(CS), BS(CS), dsc, msce, doc, postgraduate, undergraduate, MSE, bsit, MD
Research Interest	abstract state machine, adaptive system, algorithm, ambient intelligence, analytical database, antivirus software, applied statistics, artificial immune, artificial intelligence, artificial life, assembly language, association rule, at model, automata theory, automated deduction, automated theorem proving, autonomous system, awareness, axiomatic semantics, bayesian network, behavior based robotic, behavioral experiment, binary decision diagram, bioinformatics, bionics, boolean algebra, brain imaging, categorical sequence, chemical computer, children, cholesky decomposition, classification, classification algorithm ,cloud computing, cluster analysis, cluster computing

3.4 Proposed Methodology

The proposed methodology will address following research questions.

3.4.1 Research Question 1

How accurately can we extract schema/data from web lists dealing with their inherent variations?

After applying algorithm 1 and algorithm 2, we came to know that we can extract schema and data of web lists. The results and evaluation for these algorithms have been discussed in chapter 4 section reference.

Most of the existing techniques extract table schemas of two dimensional tables in which headings are usually represented in first row while remaining rows act as data rows. Hence, tables' schema is extracted from first row. However, schema extraction from lists is different than schema extraction from table. In case of web lists, schema and data extraction is performed the same way because of intermixing of schema and data values. Following algorithm is used to extract schema and data from web list.

Algorithm 1: Extract Schema and Data

Input: HTML source code
Output: Data Records with column and row separators

Begin
 S=NULL
 A[]=NULL
 for (line = first line to last line)
 S= RemoveTags(line)
 A<-add(S)
 A<-add("Next Record" string as row separator)
 End for
End

Procedure: RemoveTags(t)

Input: HTML code line
Output: string with column delimiter

Begin:
S'=NULL
 while(!end of line)
 if(html tag)
 S1=replace html tag with '\t'
 S'=S'+S1
 else if(TEXT)
 S'=concat(S',TEXT)
Return S'

End

FIGURE 3.6: Algorithm to extract schema and data from web list.

The above algorithm 1 is applied on each input file to extract schema and data of faculty list. Both schema and data are extracted in same step.

Algorithm 2: Data Cleaning**Input:** List of data Records with column and row separators, records[]**Output:** cleaned data with row separator

Begin:

```

Cleanedrecords[]=""
j=0;
for(i=0;i<records.count;i++)
  if(!next record)
  {
    While(!TEXT)
    {
      Remove "\t "
    }

    Cleanedrecords [j++]= TEXT
  }
  else
  Cleanedrecords[j++]="next record"
End for

```

End

FIGURE 3.7: Algorithm to clean data.

Algorithm 2 is applied on output gained from algorithm 1. It removes all column separators and splits each column value into simple text string. After applying Algorithm 1 and Algorithm 2, schema and data values have been extracted.

3.4.2 Research Question 2

To what extent can schema matching be performed on schema extracted from web lists?

After applying Schema Matching algorithm, we came to know that we can match schema web lists. The results and evaluation for schema matching algorithm have been discussed in chapter 4 section reference.

As we are extracting visible text from HTML tags, so schema can be given in one tag along with its data value. It is also possible that schema is given in one tag while its data value will be in next tag if it is non-empty. It has been observed that if both schema and data value are given in one tag, then they are generally

separated using colon (:). If both attribute and data value are in one line separated using “:” then, the text string is split based on colon and first part is checked for attribute name while next part is again checked whether it is data value of that attribute, or it is some other attribute or data. Two cases for schema given in web list are possible:

1. Schema is given for all attributes.
2. Schema is partially given i.e. with some attributes it is given and for some, it is not.

Case 1: First, a text string is matched with existing attributes using name equality technique. If it does not match with existing attribute, then it is matched with attribute synonyms. If text string is matched with some existing attribute, or with its synonym, this means this text string is part of table schema. However, if all attribute names (schema for all values) are given but still there is a chance of having missing values against some attributes, due to which if an element is identified as schema (attribute name), then we cannot assign next value directly to the matched attribute. There is a need to check the data value as well.

Case 2: Same method as described in case 1 will be used for this case too. If text string is matched with some existing attribute, or with its synonym, this means this text string is part of table schema. But this information is not enough to classify the next text string as data value of matched attribute because next several text strings can be the data values of existing attribute. See Figure 3.8 (a), and 3.8 (b), where two and four values against attributes “Email” and “Qualification” respectively are given.

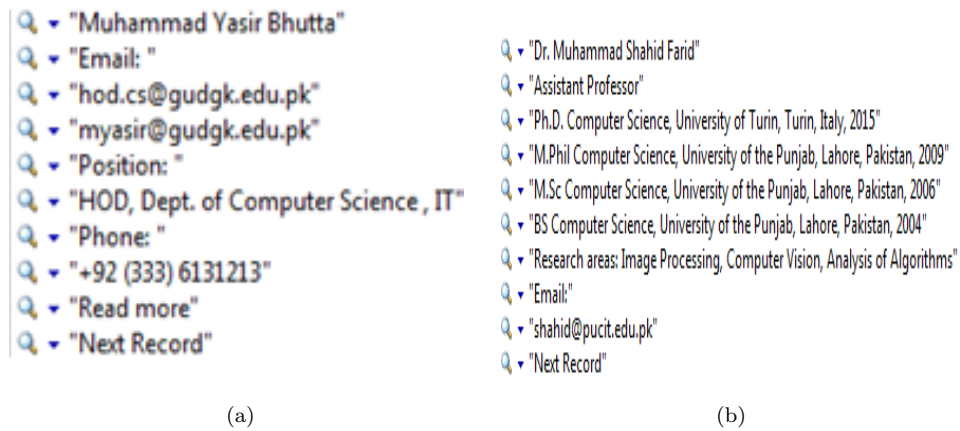


FIGURE 3.8: (a): Multiple values for Email, (b): Multiple values for Qualification.

It is also possible that only attribute name is present but data value for that attribute on next line is missing. The next value may be some other attribute or data value of some other missing. See Figure 3.9 (a) and Figure 3.9 (b) for such examples.

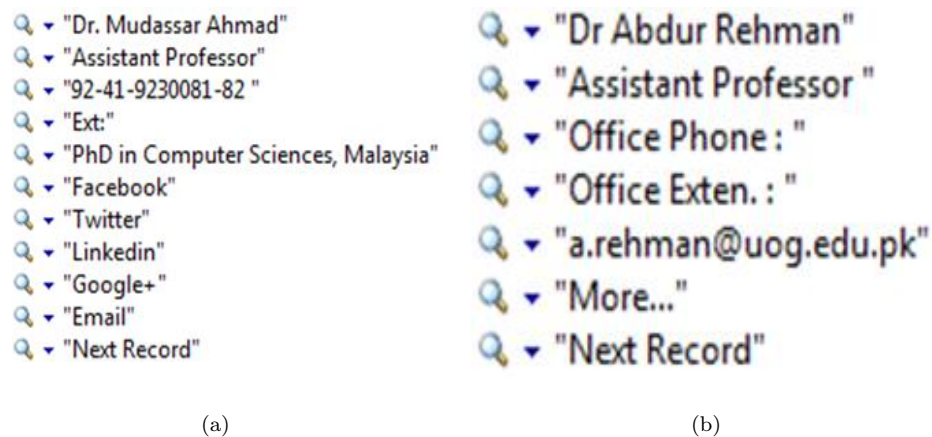


FIGURE 3.9: (a): Missing values, (b): Missing values.

Following schema matching algorithm has been applied to match schema with attributes names of target table and synonyms of that attribute.

Algorithm 3: Schema Matching

```

Input: List element      [Data Value (attribute value)]
Output: return true if match is successful, false otherwise
Begin:
    Remove all punctuation marks from data value to get a string of alphabets and numbers
    Extract attributes names from target table
    Match text string with each attribute name using equal string matching technique
    If match is successful
        return true
    else|
        Match string with entries of "Attribute_Synonym" table using equal string matching technique
        If match is successful
            return true
        else
            return false

return
End

```

FIGURE 3.10: Schema Matching Algorithm.

The result of this algorithm has been described in Chapter 4.

3.4.3 Research Question 3

Can data extracted from web lists be integrated into database table?

After applying instance based matching algorithm, we came to know that we can extract data of web list and assign it to its respective attribute. The results and evaluation for this algorithm has been discussed in chapter 4 section reference.

For data values of web list, instance based matching technique will be used to classify data values to their related attributes. Due to heterogeneity issues, mapping data values to corresponding/related attribute in global table is a hard task. If Schema Matching algorithm returns false, the text string will be passed to Data Matcher.

3.4.3.1 Data Matching/Data Classification

For instance matching, following methodology has been used for each attribute value. Attributes are classified as structure based or taxonomy based and handled

Algorithm: Instance Matching

```

Input: List [] an array of n Faculty Records, Designation_Taxonomy [], Qualification_Taxonomy [],
Research_Interest_Taxonomy []
Output: A relational table
Begin:
List []
I=NULL
for (I=start of list to I=end of list)
    While(list[i]! ="Next Record")
        Remove all special characters from list[i]
        Convert list[i] to Title Case
    If(list[i] is a column heading)
        move to next element to find data element
    If (list[i] contains both attribute and data)
        Split attribute and value part and check for heading or data
        If (! Heading)
            Perform all steps in else part
    else
        If (classified VALUE contains PERSON)
            Store value of list[i] in Name field
        If (list[i] contains "@" or "AT")
            Store in email field
        If (list[i] contains only numbers and numbers<=7)
            Store in phone no field
        If (list[i] contains only numbers and numbers>7)
            Store in Extension field
        If (list[i] matches with Designation Taxonomy)
            Store list[i] in designation field
        If (list[i] matches with Qualification Taxonomy)
            Store list[i] in Qualification field
        If (list[i] matches with Research Interest Taxonomy)
            Store list[i] in Research Interest field
        If (list[i] matches with Research Interest Taxonomy and list [i-1] equals specialization)
            Store list[i] in Specialization field
        If (list[i] contains "leave" or list[i] contains " job" or list[i] contains "active")
            Store in Job Status

```

FIGURE 3.11: Instance Based Matching Algorithm.

accordingly. Attribute name is not classified in any of above category as classifier classifies is used for its classification.

3.4.3.2 Name

In order to classify a data value as Faculty Name, Stanford NLP library Stanford.NLP.NER.3.7.0.1 has been used which is a Java implementation of Name Entity Recognizer. The English model "english.all.3class.distsim.crf.ser" has been used which classifies a text string as one of three classes i.e. PERSON, ORGANIZATION, AND Location. Names written in capital or small letters affect the accuracy of the classifier. So, in order to cope with this issue, all names are first converted to title case and then given to the classifier.

3.4.3.3 Attributes Classified Based on Taxonomy

Designation Taxonomy of all possible designations has been created. See Table 3.2 for designation taxonomy. Data values are matched with this taxonomy and if it is matched then it is classified as designation. It has been observed that one keyword is same in many designations, so for those designations only same keyword is stored in taxonomy to reduce the number of terms in taxonomy table. For example, in professor, assistant professor, associate professor, distinguishing professor; only keyword professor has been stored in taxonomy to lessen the size of taxonomy table. It is also possible that on some web sites designations are written in uppercase or Initcaps, while in other websites, lower case has been used. In order to cope with letter cases, all designations are stored in lower case in taxonomy table. When a text string is matched with taxonomy table, it is temporarily converted to lower case letters for the purpose of matching with designation taxonomy. However, in case of true match, original text string is stored in faculty table against designation attribute.

Department As Faculty data of Computer Science department has been used in this research, and most of the websites do not embed department information along with each data record, so, if it is provided explicitly with faculty records, then it is added in department field, otherwise, value “Computer Science” is added in department field. For example, “Faculty - Computer Science”, “Department of Computer Science”, “dept CS”, etc.

The keywords “department”, “dept”, “faculty”, “school” and “section” has been used to classify a value to department field. However, it is also possible that these keywords may be used in some other fields such as “Adjunct Faculty”, “Department Coordinator”, “Department Head”. They are designation values but contain keyword for department field. In order to handle such issues, if a value against above keywords is matched, then before assigning it to department field, first it is checked that if the part of string does not belong to designation or qualification filed. If so, then it is added to department field. Otherwise it is added to other related attribute based on next condition matching.

Qualification

Taxonomy of all possible official degree titles has been created. Data values are matched with this taxonomy. If part of text string contains a value which matches with this taxonomy, then it is classified as qualification. Due to heterogeneity issues, one degree title may be written on different web pages differently, for example degree title “PHD” can be written in following different ways such as “Ph.D.”, “PhD”, “Ph.D”, “Ph. D”, “phd”. Similarly, degree title “MS” can be written in many different ways such as MS(CS), MS CS, M.Phil Computer Science, MS Computer Science, MS in computer Science, MSCS, M.Phil (CS), M.S. (Computer Science), MS-Computer Science.

In order to handle letter case (upper case, lower case, initcap), all text strings are converted to lower case and in order to handle different punctuation marks, all punctuation marks have been removed to make match process simple. However, in target table, actual text strings are stored. As part of text string is matched with qualification taxonomy, and some degree titles such as “MS”, “MED” etc can be part of other terms. For example, if “ms” and “med” in lower case is stored as degree title, then research interest values “systems” and “multimedia” also contains these degree terms. So, terms with two and three letters are left uppercase while terms with more than two characters are stored in lower case in qualification taxonomy.

An issue arises when e.g. for the following cases, BS(CSE) Islamia University, Bahawalpur, BS (Computer Sciences) FAST-NUCES, Karachi, MSE(Software Engineering) COMSATS, Islamabad, qualification is described using one tag and institute from where qualification has been achieved is in another tag, then second value does not contain any degree word specified in qualification taxonomy. However, this value is also part of qualification. So, these values are concatenated with qualification field using condition that previous value is in qualification field and current value either contains university keyword or Location.

Research Interests

Taxonomy of research topics in the area of Computer Science has been created in database. For this, Research topics in ACM and IEEE dataset have been extracted from study conducted by Hoonlor (Hoonlor et al., 2012). These research interest are then stored in database table. The list has been split based on comma. After splitting, list has been sorted alphabetically; duplicate research areas have been removed. After removing duplicate research areas, a total of 552 research topics were extracted and stored in taxonomy table.

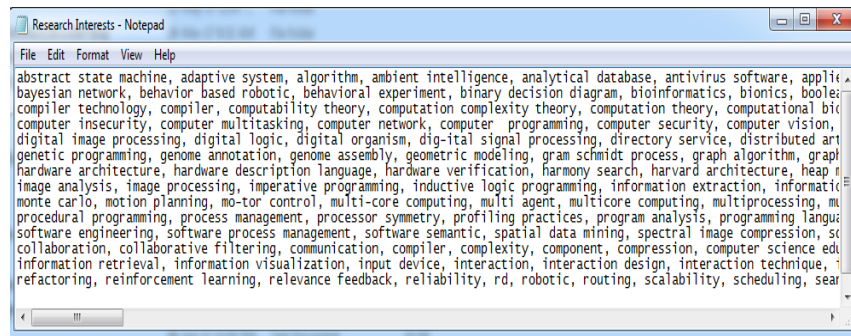


FIGURE 3.12: Research areas extracted from work of Hoonlor (Hoonlor et al., 2012).

The Figure 3.13 shows snapshot of research interest stored in Taxonomy table.

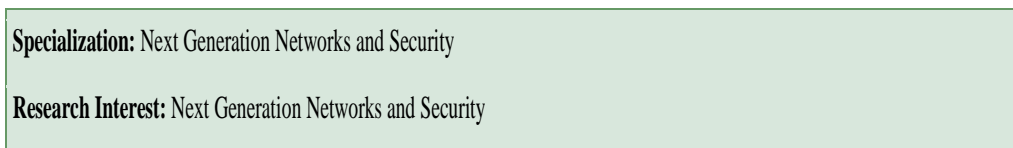
Rinterest	
1	abstract state machine
2	adaptive system
3	algorithm
4	ambient intelligence
5	analytical database
6	antivirus software
7	applied statistics
8	artificial immune
9	artificial intelligence
10	artificial life
11	assembly language
12	association rule
13	at model
14	automata theory
15	automated deduction
16	automated theorem p...
17	autonomous system
18	awareness
19	axiomatic semantics
20	bayesian network
21	behavior based robotic
22	behavioral experiment

FIGURE 3.13: Taxonomy of Research Interest in Taxonomy table.

First a text string is matched with research interests taxonomy, if part of string matches with research interest taxonomy, then text string is assigned to attribute Research Interest, otherwise taxonomy of research interest is split on the basis of space character and individual terms are obtained. Text string is then compared to these individual terms, if match is found, text string is classified as research interest.

Specialization

In some websites, only research interests are given, while on some specialization and both research interests are given. In some websites, research interest and specialization both fields contain same data e.g. see Figure 3.14. So, in case when no heading is specified, it cannot be predicted whether it is research interest or specialization, so it has been observed that for specialization filed, heading “Specialization:” is generally mentioned. Based on this observation, it is assumed that if research topics are specified without heading, then this is the value of research interest filed. However, if current value matches with research interest, and previous value is Specialization, then current value belongs to Specialization field.



Specialization: Next Generation Networks and Security
Research Interest: Next Generation Networks and Security

FIGURE 3.14: A fragment of faculty record with same specialization and research interest.

3.4.3.4 Attributes Classified Based on Structure

Job Status

Title of job status is generally not present on websites. When showing job status of faculty members, text string such as “On Duty”, “On Leave”, “On Study Leave”, “On Extraordinary Leave” is generally written. So, any such value is classified

as Job Status value, which is matched with keywords “leave”, “duty”, “job” or “sabbatical”.

Email

As e-mail id of a person generally contains a character '@', so data values containing such character are classified as Email. But in some cases this character is missing, instead some other formats such as given below are also possible, canas 'at' wfu.edu, ahalt (at) cs.unc.edu, imran.farid AT pucit DOT edu DOT pk

So, in order to cover more websites, email addresses have been extracted based on match of '@', "DOT", "'at'", "(at)". Only "at" can't be used for matching because it is common to be used in other terms too e.g. sabbatical contains substring "at". So, this may be incorrectly classified as email.

Phone No and Office Extension

If a text string contains only numbers and its length is less than or equal to seven, then it is classified as Office Extension otherwise it is classified as Phone No. It is possible that phone number and extension are written in single line using one HTML tag as: Phone: +92-051-5467856 ext:354 or +92-51-9272614 (ext. 231) On Duty. For these kinds of formats, text string has been checked for containing possible occurrence of Phone no, or office extension attributes and their synonyms. If text string contains it, then text and numbers part are separated. Both Office Extension and Phone No are separately stored. Then their length has been checked and length less than or equal to seven is classified as office extension while length greater than seven is classified as phone no.

Web Profile

In some websites, detailed page is accessed with Form buttons, and hence its URL is not available in HTML source code, some pages have details in .aspx format and hence it cannot be accessed via search engines. Only detailed pages with http hyperlink available in source are extracted and stored in database. Some

websites have web page address of their detailed profile as a value on web page (example:IIU) which is considered as schema.

Room No

Data values in field “Room No” are generally alphanumeric as it is generally made up of as building block number and room no. It has been observed that in faculty domain heading of room numbers on some faculty web pages is given while missing in other web pages. If text string is not classified as value of any attribute of target table, then its previous value is checked, if previous value contains Room or Office attribute, then current value is assigned to Room attribute.

3.4.3.5 Heterogeneity Issues

Following is a list of different heterogeneity issues for above attributes in the domain of Faculty data.

TABLE 3.3: Heterogeneity Issues in the domain of Universities’ Faculty.

Attribute	Schema Conflicts	Data Value Conflicts
Name	Faculty Name, Fact_Name, F. Name, First Name, Last Name	One example: Salahuddin, Salah Ud Din, Salah-ud-din
Designation	Position: Designation:	HOD, Head of Department. Principal Chair, chairperson, chairman, Incharge, Chairperson Teaching Fellow, Teaching Assistant, Co-operative Teacher
Department	Department:	School, Dept., department, dept, section, faculty
Qualification	Qualification, Qualifications	“Ph.D.”, “Ph.D.”, “Ph.D.”, “phD”, “ph.D”

Attribute	Schema Conflicts	Data Value Conflicts
Email	Email, Email, E, e-mail, E-mail, Email:	imran.farid AT pucit DOT edu DOT pk, V.Hall.1@warwick.ac.uk, canas 'at' wfu.edu, ahalt (at) cs.unc.edu
Phone No	Office Phone:, Phone No: , Phone no, Pno, Phone#, T:, Tel:, Telephone: , Ph., Cell:	0092 (53) 3040223, +92 (848) 550275, 203.432.4712, 203-432-4091
Extension	Office Exten:, Ex, Ex#, ext: , Ext:, ex_210	ex_210
Job Status	Status	On Study Leave, On Study Leave for PHD, On Job, Active, On Duty
Research Interest	research interest, research interests, research areas, area of interest, Research Interests:, Expertise:, Area of expertise:, Interests:	
Web Profile	Profile, more, read more, view profile, profile, Show Details, View Detail Profile, Web:, View Home Page, Homepage, Personal Homepage,	
Room	Office, O:, Room:,	

3.4.3.6 Semi-Automated Approach

Semi-automatic approach involves administrator to classify an unclassified value to its associated attribute or create it as new attribute or ignore it as garbage

value.

3.4.3.7 Unmatched/Unclassified Text String

If a text string does not match with existing attribute, attribute synonyms, and it is also not classified by instance based matching then the text string may be a new attribute, a or synonym of an existing attribute or unclassified data value. The decision of classifying the unclassified text string into above mentioned types will be made by the Administrator. Following window will open for unclassified text string.

Input Form

List of Existing Attributes

- Sno
- Name
- Designation
- Department
- Dept_id
- Uni_id
- Campus_id
- Qualification
- Email
- PhoneNo
- Fax
- Specialization
- Research_Intere
- Room_No
- Office_Extensio
- Job_Status
- Web_Profile
- Semester
- Campus

Instructions:

1. If unclassified value belongs to the any of existing attributes. Just select the corresponding attribute and Click on "Insert Data" button.
2. If it is synonym of one of already existing attributes then select the attribute from given set and click on "Add as Attribute Synonym" button.
3. If it is a new attribute. click on "Add New Attribute" button.
4. If it is new Designation, Qualification or Research Interest term, then select relevant attribute and click on "Add in Taxonomy" button.
5. If it is garbage value, Click on "Ignore" button.

Buttons: Insert Data, Ignore, Add as Attribute Synonym, Add New Attribute, Add in Taxonomy, Close

FIGURE 3.15: Input Form for Administrator.

If it is a data value of some existing attribute, then administrator will select an attribute from set of existing attributes and then click on “Insert Data” button. The data value will be added against selected attribute in “Faculty” table. If it is synonym of some existing attribute, then administrator will select the relevant attribute and then click on “Add as Attribute Synonym” button. The text string will be added in “Attribute_Synonym” table. If it is a new attribute, administrator will click “Add New Attribute” button and following window will open which will ask administrator to enter attribute name and code. New attribute will be added in Faculty table. This new attribute and its code will also be added in “attributes” table.

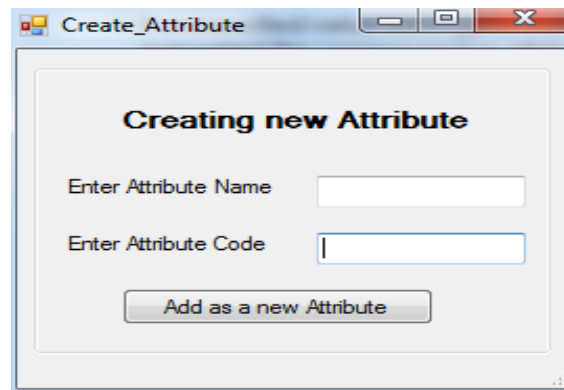


FIGURE 3.16: Input Form for Administrator.

While extracting text from HTML tags, it is possible to also have such data which is not of our interest. We will call such text as garbage value. In case of garbage values e.g. in following case `<div>` tag has not been closed properly and hence has become data part. All such kind of values which are related to Faculty domain are ignored. As this value may be repeated with each faculty record so it will be added to list of ignored value. Once it becomes part of ignored list, then on each occurrence of this value, it will be automatically ignored. Administrator will not be asked again and again for its classification.

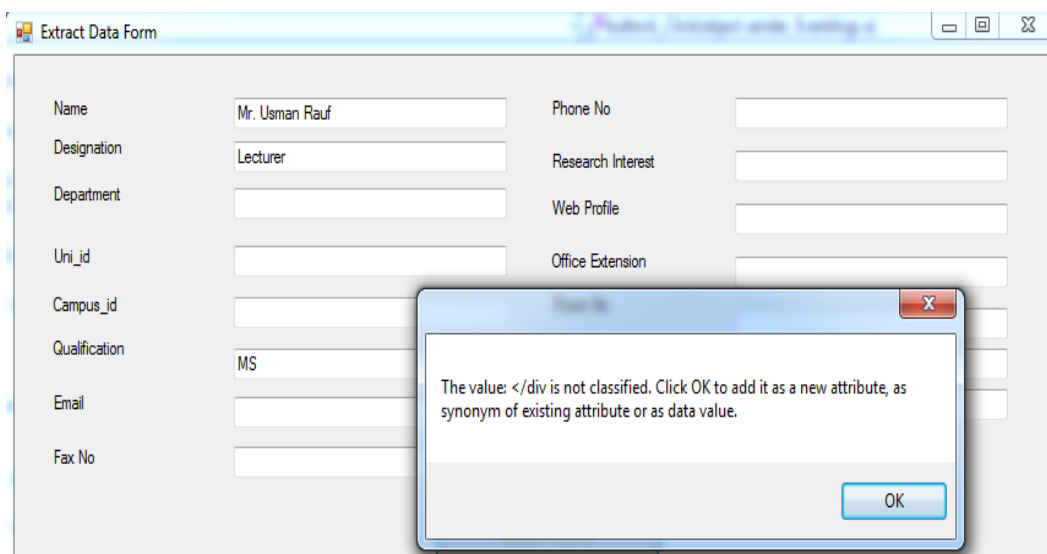
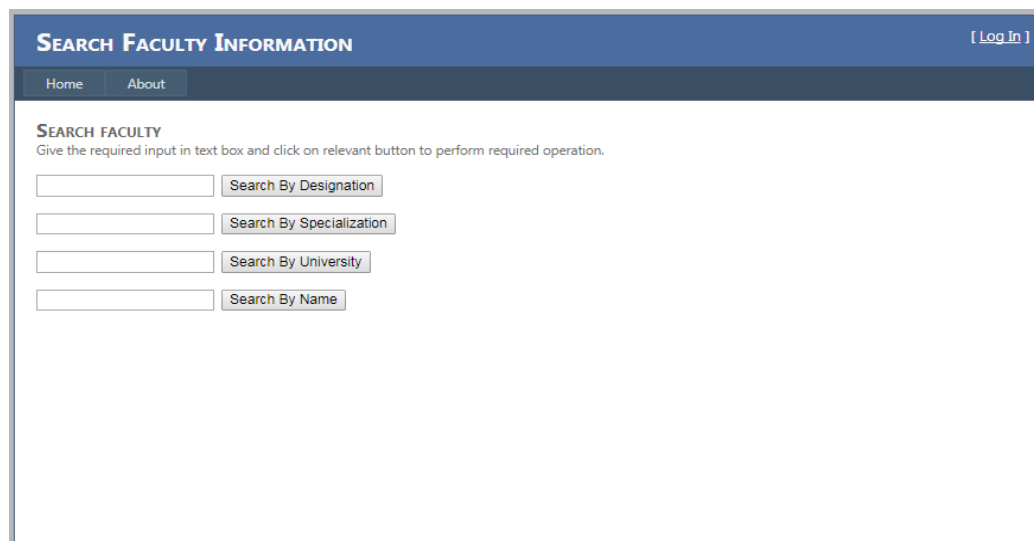


FIGURE 3.17: Garbage Value.

3.4.3.8 Faculty Search-A Web Application

A web application in asp.net has been developed which can be used to search faculty information of 110 universities which have been stored in database. The Figure 3.18 shows interface of the web based application. Users of this application can search required information by designation, specialization, university and by teacher name. They only have to provide required input in input boxes and click on relevant button to search for required information.



The screenshot shows a web application interface for searching faculty information. The page has a blue header with the title "SEARCH FACULTY INFORMATION" and a "[Log In]" link. Below the header is a navigation bar with "Home" and "About" links. The main content area is titled "SEARCH FACULTY" and contains the instruction "Give the required input in text box and click on relevant button to perform required operation." There are four search options, each with a text input box and a button: "Search By Designation", "Search By Specialization", "Search By University", and "Search By Name".

FIGURE 3.18: Interface of Webpage of Search Faculty Information.

The Figure 3.19 shows the output by selecting designation as “Lecturer”. Here, due to space limitation, few records have been shown.

Figure 3.20 shows the extracted records searched by specialization “Artificial Intelligence”.

Name	Designation	Email	PhoneNo
Ms. Atifa Sarwar	Lecturer	atifa.sarwar@nu.edu.pk	
Shahbaz Ahmed Alvi	Lecturer, Mathematics		
Yousuf Kerai	Lecturer, Mathematics		
Tariq Mumtaz	Lecturer, Electrical Engineering		
Muhammad Jabbar	Lecturer	jabbar_uet47@yahoo.com	
Muhammad Sami Ullah	Lecturer	msamiullah@uog.edu.pk	+92 (53) 3643112
Muhammad Tasaddaq Latif	Lecturer	mtasaddaq@yahoo.com	
Adeel Ahmed	Lecturer	adeelahmed292@gmail.com	
Anbreen Kausar	Lecturer	anbreen.kausar@uog.edu.pk	
Mrs Ayesha Altaf	Lecturer	ayesha.altaf@uog.edu.pk	
Ehtisham Rashid	Lecturer	onlyehtisham@yahoo.com	
Fakhra Nazir	Lecturer	fakhra.nazir@uog.edu.pk	
Hafiza Basserat Fatima	Lecturer	basserat.fatima@uog.edu.pk	
Muhammad Abo Bakar Aslam	Lecturer	abobakar@live.co.uk	
Mr Muhammad Arif	Lecturer	arifmuhammad36@hotmail.com	
Muhammad Bilal Ahmad Janjooa	Lecturer	bilal.janjooa@uog.edu.pk	
Saleem Afzal	Lecturer		0533-643216

FIGURE 3.19: A Fragment of Records Searched by Designation (lecturer).

Name	Designation	Email	Qualification	Research Interest
Dr. Waqar ul Qounain	Assistant Professor	swjaffry@pucit.edu.pk	PhD Computer Science (Vrije University, Amsterdam, The Netherlands), M.Sc Computer Science (University of The Punjab, Pakistan), PGD Computer Science (University of The Punjab, Pakistan)	Computational Modelling, Artificial Intelligence, Data Mining and Machine Learning, Multi-Agent Systems,
Dr. Fawad Hussain	Associate Professor (HEC Approved PhD Supervisor)	fawadhussain@giki.edu.pk	PhD in Machine Learning (Grenoble, France); MS Computer Science (Paris, France);	Machine Learning, Big Data Analysis, Data Mining, Artificial Intelligence, Semantic Analysis,
Dr. Imran Amin	Associate Professor , Head of Computer Science Department		Ph.D(Loughborough, UK)	Embedded Systems, Artificial Intelligence and Image Processing,
Runhe HUANG				Artificial intelligence, Machine learning, Neural network, Data mining and knowledge fusion, Knowledge representation and configuration, Knowledge discovery and fusion, Human cognitive process modeling, Associative memory and recall modeling,

FIGURE 3.20: A Fragment of Records Searched by Specialization/Research Area as Artificial Intelligence.

Figure 3.21 shows the extracted records searched by university “Virtual University”.

Name	Designation	Qualification	Department	University
Asma Batool	Assistant Professor	MS Computer Science	Department of Computer Science	Virtual University of Pakistan
Dr. Muhammad Tariq Pervez	Assistant Professor	Ph.D. (Bioinformatics)	Department of Computer Science	Virtual University of Pakistan
Dr. Nasir Naveed	Assistant Professor	PhD (Computer Science)	Department of Computer Science	Virtual University of Pakistan
Hasnain Ahmed	Assistant Professor	MS Computer Science	Department of Computer Science	Virtual University of Pakistan
Muhammad Anwaar Saeed	Assistant Professor	M.Phil Computer Science	Department of Computer Science	Virtual University of Pakistan
Muhammad Jawwad Zaheer	Assistant Professor	Ph.D (In Progress), MS (Computer Science)	Department of Computer Science	Virtual University of Pakistan
Muhammad Salman Bashir	Assistant Professor	MS Computer Science	Department of Computer Science	Virtual University of Pakistan
Muhammad Summair Raza	Assistant Professor	MS (Software Engineering)	Department of Computer Science	Virtual University of Pakistan

FIGURE 3.21: fragment of records searched by University.

Figure 3.22 shows the extracted records searched by university “Name”.

SEARCH FACULTY INFORMATION [Log In]

[Home](#) [About](#)

SEARCH FACULTY
Give the required input in text box and click on relevant button to perform required operation.

Name	Designation	Qualification	Email	University	Research_Interest	Job_Status
CHANDIO SHAHMURAD	Lecturer	BSIT. (S.U) 2005	sm.chandio@usindh.edu.pk	University of Sindh		on study leave

FIGURE 3.22: A fragment of extracted records by Name.

In this chapter, overall methodology to conduct this research has been described. Four algorithms have been devised to carry out this research. Algorithm 1 and algorithm 2 extract schema and data from web list. Schema matching algorithm matches attributes with attributes of target table and its synonyms. Instance based matcher is applied on extracted data to classify it to its respective attribute. If attributes and data values are not matched and classified, then semi-automated technique is used which involves administrator to identify it as new attribute or assign it as data value to some attribute. Results and evaluation of these algorithms have been discussed in Chapter 4.

Chapter 4

Results and Evaluation

Experiments according to methodology described in chapter 3 are performed on 110 websites which belong to the domain of Faculty data. Algorithms 1 and 2 have been applied to extract data and schema from web lists. Algorithm 3 has been applied for the task of schema matching. In last step, algorithm 4 has been applied which performs instance level matching on data extracted from web lists. This instance level matching classifies each data value to its corresponding attribute and then stores it in database table hence performing the step of data integration.

In order to measure the performance of data extraction algorithms, standard measures precision and recall are heavily used (Liu et al, 2006). In this research, three types of evaluation methods have been used to assess the performance of proposed technique. The evaluation methods include the following.

1. Quantitative Analysis
2. Comparison with existing approaches
3. Query Based Validation

The proposed technique has been implemented in two ways i.e. fully automated and semi-automated way. The automated approach does not involve administrator intervention, however, semi-automated approach involves administrator to decide

for unmatched attributes and data values. The results presented below are given for automated approach. In case of semi-automatic approach, all unclassified data will be classified to their respective attributes by the administrator. For each unclassified value, administrator will be asked to decide whether to ignore this value or add as data value in the integrated table or add as attribute synonym etc. The detail of this process has been described in Chapter 3.

4.1 Distribution of Attributes

The graph given below shows the distribution of attribute instances that are present of 110 websites.

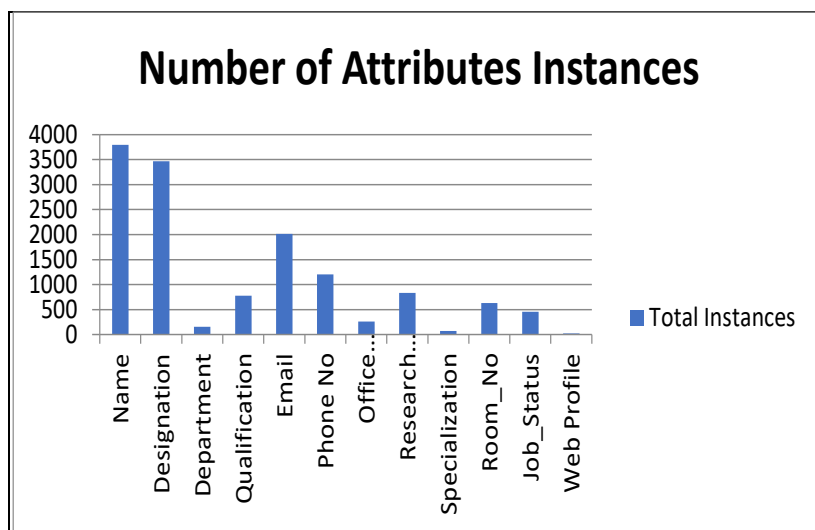


FIGURE 4.1: A Number of Attributes Instances on 110 websites.

The graph shows that faculty name is present on all faculty web pages; the second largest used attribute on most of the website is Designation. Email and Phone No attributes are also heavily used by many websites.

4.2 Quantitative Analysis

Following formulas have been used to calculate precision, recall and F-measure of each attributes.

$$Precision = \frac{CorrectlyExtracted(TP)}{CorrectlyExtracted(TP) + IncorrectlyExtracted(FP)} \quad (4.1)$$

$$Recall = \frac{CorrectlyExtracted(TP)}{CorrectlyExtracted(TP) + NotExtracted(FN)} \quad (4.2)$$

Where, TP , TN , FP , and FN stand for True Positive, True Negative, False Positive, and False Negative, respectively.

$$F - Measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (4.3)$$

4.3 Results of Research Questions

After experimentation step, following three research questions have been addressed. Detail of the answer of each research question is described below.

4.3.1 Research Question 1

How accurately can we extract schema/data from web lists dealing with their inherent variations?

First of all, algorithm 1 has been applied on each input file. Following diagram shows the output fragment of running Algorithm1 on list of Ghazi University (GU), Pakistan.

```

Q ▶ "t t t t t t t t t t t t t t t t t t t t Muhammad Jasim Shah t t t t t t t t Email: t t t t t t mjasim@gudgk.edu.pk t t t t t t t t Position: t t t t t t Lecturer t t t t t t t t t t Read more t t t t t t t t t t"
Q ▶ "Next Record"
Q ▶ "t t t t t t t t t t t t t t t t t t t t Muhammad Gulzar t t t t t t t t Email: t t t t t t mgulzar@gudgk.edu.pk t t t t t t t t Position: t t t t t t Lecturer t t t t t t t t t t Read more t t t t t t t t t t"
Q ▶ "Next Record"
Q ▶ "t t t t t t t t t t t t t t t t t t t t Zaib-ul-Nisa Khosa t t t t t t t t Email: t t t t t t tzkhosa@gudgk.edu.pk t t t t t t t t Position: t t t t t t Lecturer t t t t t t t t t t Read more t t t t t t t t t t"
Q ▶ "Next Record"

```

FIGURE 4.2: Output Fragment of Applying Algorithm1 on GU List.

In next step, Algorithm 2 is applied on output gained from algorithm 1. It removes all column separators and splits each column value into simple text string as follows.

```

Q ▶ "Muhammad Jasim Shah"
Q ▶ "Email: "
Q ▶ "mjasim@gudgk.edu.pk"
Q ▶ "Position: "
Q ▶ "Lecturer"
Q ▶ "Read more"
Q ▶ "Next Record"
Q ▶ "Muhammad Gulzar "
Q ▶ "Email: "
Q ▶ "mgulzar@gudgk.edu.pk"
Q ▶ "Position: "
Q ▶ "Lecturer"
Q ▶ "Read more"
Q ▶ "Next Record"

```

FIGURE 4.3: Output Fragment of Applying Algorithm2 on Output Of Algorithm 1.

After applying Algorithm 1 and Algorithm 2, schema and data values have been extracted. The results have been evaluated in following different ways:

In first method, 20% websites have been randomly selected and algorithms 1 and 2 have been applied on these websites. The obtained results are manually checked and after manual verification, it is found that results of algorithm 1 and 2 are 100% accurate.

As next phases of schema matching and data integration depends on the results of algorithm 1 and 2, so their success (accuracy) is reflecting the success of this phase too. So, we can determine the accuracy of this phase from the accuracy of subsequent steps too.

The Precision, Recall and F-measure of ListExtract technique (Elmeleegy et.al, 2009) is 64%, 63% and 63% respectively. This technique is fully automated and precision, recall and F-measure of our technique is higher than ListExtract because we are creating input file of source code using Google chrome extension: Advanced Web Scrapper, which is only extracting only records of interest, hence generally no irrelevant code is extracted.

4.3.2 Research Question 2

To what extent can schema matching be performed on schema extracted from web lists?

This step is based on two sub steps which are schema identification and separating it from data. To evaluate schema matching algorithm, it has been checked manually whether schema given on a website is actually matched with schema of global integrated table. For this, as a test data of 10 more websites in Computer Science faculty domain is collected and proposed technique is applied on those websites. The schema identified by proposed approach has been stored in temporary file and matched with schema present on web site. Then, performance measures, Precision, recall and F-measure have been calculated to assess the performance of schema matching algorithm. The Table 4.1 shows the calculation of Precision, Recall and F-measure of proposed schema matching algorithm. TP (True Positive) is the number of attributes correctly identified as schema, FP (False Positive) is the number of attributes that are incorrectly identified as schema, and however, actually it was either some data value or some garbage value. FN (False Negative) shows the number of attributes that are not extracted i.e. they are not identified as schema.

TABLE 4.1: Precision, Recall and F-Measure of Instance Matching Algorithm.

Uni Short Name	Total Records on Web Page	No of attributes /Schema	Correctly Extracted	Incorrectly Extracted	Not Extracted	Precision	Recall	F-Measure
vt	57	267	228	0	39	1	0.85	0.92
uab	10	20	10	0	10	1	0.50	0.66
oldcs	19	394	94	0	0	1	1	1
metu	34	169	135	0	34	1	0.79	0.88
usf	45	40	0	0	40	0	0	0
uh	58	40	40	0	0	1	1	1
fsktm	129	1032	774	0	258	1	0.75	0.85
Aut	52	156	104	0	52	1	0.66	0.80
infolab	9	45	27	0	18	1	0.60	0.75
kaust	16	16	0	0	16	0	0	0

Table 4.1 shows Precision, Recall, and F-measure of all ten websites. Then Average Precision, Average Recall, and Average F-Measure have been calculated. The results show that for test set of 10 websites, average precision, average recall, and average F-measure of proposed schema matching algorithm is 80%, 62% and 69% respectively.

Average Precision = 80%

Average Recall = 62%

Average F-Measure = 69%

Another evaluation is performed in which a random sample of 20% websites have been chosen from dataset of 110 websites and schema identified by proposed approach has been stored in temporary file and matched with schema present on website. Table 4.2 shows Precision, Recall and F-measure of these websites.

TABLE 4.2: Precision, Recall and F-Measure of Schema Matching Algorithm.

Uni Name	Total Records on Web Page	No of Attributes	TP	FP	FN	Precision	Recall	F-Measure
BKU	14	70	70	0	0	1.00	1.00	1.00
HITEC	19	77	77	0	0	1.00	1.00	1.00
IMCS	33	33	33	0	0	1.00	1.00	1.00
CUSIT	21	21	21	0	0	1.00	1.00	1.00
COMSATS2	64	257	254	0	3	1.00	0.99	0.99
warwick	72	210	210	0	0	1.00	1.00	1.00
Uwo	24	42	20	0	22	1.00	0.48	0.65
Aub	10	48	48	0	0	1.00	1.00	1.00
kingston	11	11	11	0	0	1.00	1.00	1.00
NYU	49	146	146	0	0	1.00	1.00	1.00
dartmouth	21	21	21	0	0	1.00	1.00	1.00
GIKI	16	48	48	0	0	1.00	1.00	1.00
UOH	15	45	45	0	0	1.00	1.00	1.00
rochester	23	43	43	0	0	1.00	1.00	1.00
carleton	33	33	33	0	0	1.00	1.00	1.00

Average Precision = 100%

Average Recall = 96%

Average F-Measure = 98%

In case of 10 more websites, Precision, Recall and F-measure is less because in new dataset, some new attribute synonyms such as Education, Voice, Publication Statistics, CV, Research Keys etc are found which were not part of our attribute synonym table.

4.3.3 Research Question 3

Can data extracted from web lists be integrated into database table?

Instance based matching algorithm has been applied on data extracted from web lists. Precision, recall and F-measure have been calculated for each of the attribute, then average Precision, Average Recall and Average F-measure have been calculated.

Total attributes = 12

Total records in integrated table = 3798

Total Cells in integrated table = 45576

Filled Cells = 13457

Empty Cells = 32119

Table 4.3 shows the results of algorithm 3.

TABLE 4.3: Precision, Recall and F-Measure of Instance Matching Algorithm.

Attribute Name	Actual Values	Extracted Values	Correctly Extracted	Incorrectly Extracted	Not Extracted	Precision	Recall	F-Measure
Name	3798	3771	3473	298	27	0.92	0.99	0.95
Designation	3466	3410	3408	2	58	0.99	0.98	0.99
Department	153	152	152	0	1	1	0.99	0.99
Qualification	780	780	771	9	9	0.98	0.98	0.98
Email	2014	1970	1956	14	58	0.99	0.97	0.98
Phone No	1206	1224	1205	19	1	0.98	0.99	0.99
Office Extension	258	278	241	37	0	0.86	1	0.92
Research Interest	830	877	824	53	6	0.93	0.99	0.96
Specialization	69	69	69	0	0	1	1	1
Room_No	631	452	366	86	265	0.80	0.58	0.67
Job_Status	453	453	453	0	0	1	1	1
Web Profile	24	24	24	0	0	1	1	1

Average Precision = 0.95%

Average Recall= 95%

Average F-Measure= 95%

The following graph shows precision, recall and f-measure of all attributes.

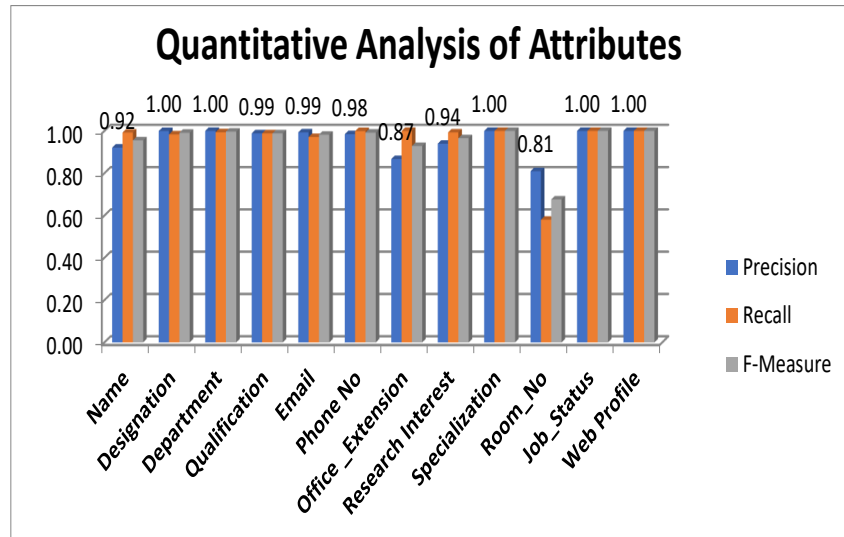


FIGURE 4.4: Quantitative Analysis of Attributes.

4.4 Query Based Validation

After integrating data extracted from web lists into global table, SQL queries can be performed on the table data. This step has been used to validate the proposed approach. Following are some queries.

4.4.1 Query 1

Following simple query has been applied on 'Faculty' table which extracts values of attributes specified in the query.

Output:

```

SELECT      Name, Designation, Email, PhoneNo, Job_Status
FROM        Faculty

```

FIGURE 4.5: Query 1.

Total rows returned by above query are 3798, however due to space limitation, only 18 rows are shown.

Name	Designation	Email	PhoneNo	Job_Status
Shoukat Ali Gil	Program Officer	shoukataligil@comsats.edu.pk	05190495044	On Duty
Dr. Nasro Min-allah	Associate Professor	nasar@mit.edu		On Study leave
Dr. Rafi Ullah	Assistant Professor	dr.rafiullah@comsats.edu.pk		On leave
Dr. Assad Abbas	Assistant Professor	assadabbas@comsats.edu.pk	03131500814	On Study leave
Qasim Arshad Choudhry	Assistant Professor	qasimac@comsats.edu.pk		On leave
Zeeshan Mehta	Assistant Professor	zeeshanmehta@comsats.edu.pk	+92-51-9235302	On leave
Shahid Hussain	Assistant Professor	shahidhussain2003@yahoo.com	00923339124427	On leave
Saif ur Rehman Khan	Lecturer	saif_rehman@comsats.edu.pk		On Study leave
Uzair Iqbal Janjua	Lecturer	uzair_iqbal@comsats.edu.pk	+92 051 9235302	On Study leave
Tahir Mustafa Madni	Lecturer	tahir_mustafa@comsats.edu.pk		On leave
Adeel Javed	Lecturer	adeel2122@hotmail.com	+92 51 8318471	On leave
Saadia Aziz	Lecturer	saadia_aziz@comsats.edu.pk	+92 51 8318471	On leave
Sajida Kalsoom	Lecturer	sajida.kalsoom@comsats.edu.pk		On leave
Nazia Hameed	Lecturer	nazia_hameed@comsats.edu.pk		On leave
Sehresh Khan	Lecturer	sehreshkhan@comsats.edu.pk		On leave
Sidra Malik	Lecturer	sidra.malik@comsats.edu.pk		On leave
Muhammad Fayez	Lecturer	muhammad_fayez@comsats.edu.pk	+92 51 8318471	On leave
Dr. Awais Ahmad	Lecturer	awais.ahmad@comsats.edu.pk		On Study leave

FIGURE 4.6: Output of Query 1.

4.4.2 Query 2

Following query has been applied on faculty data to extract such faculty members whose research area is “Database”.

```

SELECT Name, Designation, Qualification, Email, Research_Interest
FROM Faculty where Research_Interest like '%Database%';

```

FIGURE 4.7: Query 2.

Output:

Total number of records returned by this query is 40, however, for the sake of simplicity; some of them are shown in below output.

Name	Designation	Qualification	Email	Research_Interest
Mr. Usman Shehzaib	Assistant Professor		usmanshehzaib@oilahore.edu.pk	Databases, Data Mining, Big Data,
Saif Ur Rehman	(Assistant Professor)	Ph.D (In Progress), MS, MCS (Gold Medalist)	saif@uaar.edu.pk	Data Mining, Business Intelligence, Data Warehousing...
Arita Wasilewska	Associate Professor			Database mining, Bioinformatics Protein Secondary Str...
Noureen Zafar	(Lecturer)	MS Computer Science	noureen_zafar@uaar.edu.pk	Artificial Intelligence, Data mining, Image Processing, V...
Tariq Ali	(Lecturer)	Ph. D. (In Progress), MS (CS)	tariq.ali@uaar.edu.pk	Semantic Computing, Semantic Cache, Logic, Databa...
Durga Toshnival	Associate Professor		durgafec@iitr.ac.in	Data Mining and Databases, Mining Time Series, Data ...
Asif Nawaz	(Lecturer)	Ph.D Scholar, MS (Computer Science)	asif.nawaz@uaar.edu.pk	Data mining, Databases, Artificial Intelligence ,
Soon Joo Hyun	Professor		sjhyun (at) cs.kaist.ac.kr	Intelligent Database, Context Aware, Data Mining,
Chin-Wan Chung	Emeritus		chungow (at) cs.kaist.ac.kr	Database, Web, Social Network,
Fakhanda Qamar	(Lecturer)	MSCS, International Islamic University Islam...	fakhanda.qamar@uaar.edu.pk	Data Communication and Network, Advance Databas...
Tim Wieringer	Assistant Professor	Ph.D., Computer Science, University of Illinoi...		Network science, data science, machine learning, dat...
Dennis Shasha	Professor of Computer Science	Ph.D., Applied Mathematics, Harvard Univer...		shasha at cs.nyu.edu ,Network inference and protein ...
Yoon Joon Lee	Professor		yjee (at) cs.kaist.ac.kr	Database System,
Asif Sohail	Assistant Professor	M.Phil Computer Science , PUJ.M.Sc Comput...	asif@puccit.edu.pk	Databases, Data Structures, Digital Logic , Design,
Asim Rasul	Assistant Professor	MS Total Quality Management, PUJ.M.Sc Co...	asim@puccit.edu.pk	Databases, Information Systems,
Hemaapandira, Lane A.	Professor of Computer Science			lane at cs.rochester.edu ,Computational social choice (...)
Yücel Saygin	Faculty Member			Research Area:Active database systems, data mining, ...
Myoung Ho Kim	Professor		mhkim (at) cs.kaist.ac.kr	Database System, Information Processing,
Kyu-Young Whang	Emeritus, ???? ACM/IEEE Life Fellow		kywhang (at) cs.kaist.ac.kr	Database, Search Engines,

FIGURE 4.8: Output of Query 2.

4.4.3 Query 3

Following query has been applied on faculty data to extract such faculty members whose research area is “Networks” and they have designation “Assistant Professor”, “Professor” or “Associate Professor” or any other designation with occurrence of term “Professor”.

```
SELECT Name, Designation, Email, PhoneNo, Specialization, Job_Status
FROM Faculty where Specialization like '%Networks%' And Designation like '%Professor%';
```

FIGURE 4.9: Query 3.

Output:

The total numbers of records returned against above query are nine which are shown in the following output.

Name	Designation	Email	PhoneNo	Specialization	Job_Status
Robert Webber	Associate Professor	webber@csd.uwo.ca	519-661-2111	Computer Graphics, FPGAs, VHDL, Neural Networks	
Dr. Imran Ali Khan	Associate Professor	imran@ciit.net.pk	(92)992-383591-5	Wireless Networks	
Dr. Rafi-Us-Shan	Assistant Professor	shan@ciit.net.pk	(92)992-383591-5	QoS for Computer Networks, Public Safety and Secur...	
Dr. Waqas Jadoon	Assistant Professor	waqas_jadoon@ciit.net.pk	(92)992-383591-5	Machine Learning, Neural Networks, Sparse Coding/...	
Rab Nawaz Jadoon	Assistant Professor	rabnawaz@ciit.net.pk	(92)992-383591-5	Wireless Communication and Networks	
Javid Ali	Assistant Professor	javidali@ciit.net.pk	(92)992-383591-5	Computer Networks	
Dr. Osman Khalid	Assistant Professor	osman@ciit.net.pk	(92)992-383591-5	Delay Tolerant Networks, Trust and Reputation Syste...	
Faisal Rehman	Assistant Professor	frehman@ciit.net.pk	+92-300-9113346	Social Networks	In House Study Leave
Tahir Maqsood	Assistant Professor	tmaqsood@ciit.net.pk	(92)992-383591-5	Computer Networks	In House Study Leave

FIGURE 4.10: Output of Query 3.

4.4.4 Query 4

Following query with SQL join has been applied on ‘Faculty’ and ‘University’ tables and results are shown from both tables based on condition specified in the query.

```
SELECT Fac.Name, Fac.Designation, Fac.Email, Fac.Research_Interest, Uni.University_Name, Uni.Campus_City, Uni.Website_URL
FROM Faculty AS Fac
JOIN University AS Uni
ON (Fac.Uni_id = Uni.Uni_id) AND (Fac.Campus_id=Uni.Campus_id)
Where Name LIKE '%asif%';
```

FIGURE 4.11: Query 4.

Output:

Name	Designation	Email	Research_Interest	University_Name	Campus_City	Website_URL
Adnan Asif	Lecturer			Virtual University of Pakistan	Lahore	http://vu.edu.pk/AboutUs/F...
Asif Hussain	Lecturer		M.Sc.Computer Science,	Virtual University of Pakistan	Lahore	http://vu.edu.pk/AboutUs/F...
Asif Sohail	Assistant Professor	asif@puoit.edu.pk	Databases, Data Structure...	Punjab University College of Inf...	Lahore	http://puoit.edu.pk/index.ph...
Dr. Asif Khan	Assistant Professor	asifkhan@gki.edu.pk	Unmanned Aerial Vehicle (...)	Ghulam Ishaq Khan Institute	Khyber Pakhtu...	http://www.gki.edu.pk/Facu...
Asif Nawaz	(Lecturer)	asif.nawaz@uair.ed...	Data mining, Databases, A...	PIMAS-Arid Agriculture University	Rawalpindi	http://www.uair.edu.pk/uit/...
Dr. Muhammad Asif H...	Assistant Professor			National Textile University	Faisalabad	http://www.ntu.edu.pk/dcs...
Dr. Muhammad Asif U...	Assistant Professor			National Textile University	Faisalabad	http://www.ntu.edu.pk/dcs...
Raja Asif Wagan	Lecturer	raja.asif@butms.edu...		Balochistan University of Inform...	Quetta	http://www.butms.edu.pk/F...
Mr. Asif Khurshed	Assistant Web M...			ABBOTTABAD UNIVERSITY ...	ABBOTTABAD	http://web.aust.edu.pk/infor...
Saara Asif	Assistant Professor			Foman Christian College (A Ch...	Lahore	http://www.fccollege.edu.pk...
Asif Muhammad	Lecturer	asif.malk@comsats...		COMSATS Institute of Informati...	Islamabad	http://ww3.comsats.edu.pk/...
Awais Bin Asif	Research Associ...	awais@citwah.edu.pk		COMSATS Institute of Informati...	Wah	http://ww2.comsats.edu.pk/...
Muhammad Daud Ab...	Lecturer			COMSATS Institute of Informati...	Wah	http://ww2.comsats.edu.pk/...
Dr. Muhammad Wasif ...	Associate Profes...			COMSATS Institute of Informati...	Wah	http://ww2.comsats.edu.pk/...
Ms. Moizzah Asif	Research Associ...	moizzahasif@citlaho...	Social Computing and Cyb...	COMSATS Institute of Informati...	Lahore	http://lahore.comsats.edu.pk...
Mr. Asif Shahzad	Assistant Professor	asif.shahzad@citlah...	Products, Web, Software ...	COMSATS Institute of Informati...	Lahore	http://lahore.comsats.edu.pk...
Engr. Muhammad Asif...	Lecturer	asifsuryani@citlahi...		COMSATS Institute of Informati...	Sahawal	http://ww2.comsats.edu.pk/...
Asif Ali	Research Associ...	asifali@citvehari.ed...		COMSATS Institute of Informati...	Vehari	http://ww2.comsats.edu.pk/...

FIGURE 4.12: Output of Query 4.

4.5 Comparison With Existing Approaches

The proposed technique has been compared with existing approaches and this comparison has been shown in Table 4.4.

TABLE 4.4: Comparison with Existing Approaches

S. No.	Authors/ Year	Dataset Type	Schema Extraction	Data Extraction	Data Integration	Technique Used
1	Purnamasari, et al., 2015	HTML Tables	Yes	No	No	Wrapper induction based
2	Krishna, & Dattatraya, 2015	web tables and web lists	Yes	Yes	No	Unsupervised learning ap- proach, based on DOM trees and visual cues
3	Adelfio & Samet, 2013	HTML ta- bles and spreadsheets table	Yes	Yes	No	Classification technique based on supervised learning
4	Gultom et al., 2011	Web tables	Yes	Yes	Yes	DOM tree based
5	Nagy et.al., 2011	Web tables	Yes	Yes	No	Index based
6	Elmeleegy et al., 2009	Web lists	No	Yes	No	Unsupervised learning, language model and HTML table corpus
7	Gatterbauer et.al., 2007	Web tables	Yes	Yes	No	Visual box model

8	Zhai, & Liu, 2005	Web Tables	Yes	Yes	No	HTML tag tree based on Visual information
9	Embley et al., 2005	Web Tables	Yes	Yes	Yes	Ontology based
10	Wang & Lochovsky, 2003	HTML forms	Yes	Yes	No	Regular expression wrappers
11	Lerman et al., 2001	Web tables and lists	Yes	Yes	No	Unsupervised learning algorithms
12	Proposed Technique	Web Lists	Yes	Yes	Yes	Based on HTML source code

Above comparison shows that some of the techniques provide schema extraction, and data extraction of web tables. Some of them also perform data integration but few techniques provide schema and data extraction of web lists, none of the technique is found to perform data integration of web lists. The proposed technique performs schema extraction, data extraction, and data integration of web lists data.

In this chapter, results of proposed technique have been presented and evaluated using quantitative analysis, query evaluation and comparison with existing technique. In quantitative analysis, precision, recall and F-measure of proposed algorithms have been calculated. The second method to evaluate proposed technique is query validation, in which some SQL queries has been applied on integrated data and each query output is also shown. The output of applied queries shows that technique has correctly stored and integrated data. The third and last method of evaluation is comparison with existing technique. This comparison shows very few techniques are available which extracts schema and data from web lists but none of the technique has been found which performs data integration of web

lists. The proposed technique performs schema extraction, data extraction and data integration of web list data in the domain of universities Faculty.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Web lists like web tables is a rich source of information but like web tables, data of web lists is not beneficial for business communities and other users because it is scattered on different sources and it is not available at a centralized location on which queries could be performed.

In this research, a technique based on HTML source code has been proposed which stores data of web lists at a centralized location giving the advantage of applying ad-hoc queries. In first step, the proposed technique extracts data from web lists. This step extracts HTML source code of each web list, extracts text embedded between HTML tags, creates separators between data fields and values create row separator which separates one record of one faculty member from other faculty member. In this step, both scheme and data are extracted because web lists have a structure in which both schema and data are intermixed.

In next step, schema matching has been performed. The purpose of schema matching algorithm is to isolate attributes names and data values. Schema matching algorithm first compares each text value with its set of attributes in the 'Faculty' table, if no match is found then text value is compared with attributes synonyms stored in 'Synonyms' table. If match is found, text value is identified as schema.

In case of false matching, instance based matching is performed on text value. The instance based matching algorithm identifies the attribute to which this value belong. This matching is based on content of text string.

Experiments have been performed on a dataset of 110 websites in the domain of Universities' faculty. Out of these 110 web sites, 3websites have faculty data with full headings, 61 containing both values with and without headings, without headings and results have been evaluated in three different ways. First evaluation is quantitative and conducted using standard measures Precision, Recall and F-measure. Second evaluation is based on comparison with existing technique. Third is query based validation, four different queries has been applied on Faculty and University data.

Algorithm 1 and algorithm 2 have been evaluated on random sample of 20% websites from dataset of 110 websites. Precision, Recall and F-measure of algorithm 1 and 2 is 100%. For algorithm 3 (Schema Matching), 10 more websites have been collected from web and Quantitative analysis shows that on the sample of 10 websites, precision, recall and F-measure of schema matching algorithm is 80%, 62%, and 69% respectively. Instance matching algorithm has been evaluated on dataset of 110 websites. Precision, recall and F-measure of instance matching algorithm is 95%.

5.2 Future Work

1. In future, there is a need to automate the process of data collection by writing focused crawler.
2. Automating the process of extracting particular code portion from HTML web page is another area of research.
3. In this research, focused department is Computer Science but this research can be expanded to other domains as well.
4. Extracting and storing images needs to be handled.

Bibliography

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

Berlin, J. and Motro, A., 2002, May. Database schema matching using machine learning with feature selection. In *International Conference on Advanced Information Systems Engineering* (pp. 452-466). Springer, Berlin, Heidelberg.

Biskup, J., 1995, January. Achievements of relational database schema design theory revisited. In *International Workshop on Semantics in Databases* (pp. 29-54). Springer, Berlin, Heidelberg.

Blinn, A., Cohen, M.A., Lorton, M. and Stein, G.J., Blinn, Arnold, Cohen, Michael Ari, Lorton, Michael and Stein, 1999. Database schema independence. U.S. Patent 5,974,418.

Borkar, V., Deshmukh, K., & Sarawagi, S. (2001, May). Automatic segmentation of text into structured records. In *ACM SIGMOD Record* (Vol. 30, No. 2, pp. 175-186). ACM.

Cafarella, M.J., Halevy, A., Wang, D.Z., Wu, E. and Zhang, Y., 2008. Webtables: exploring the power of tables on the web. *Proceedings of the VLDB Endowment*, 1(1), pp.538-549.

Crescenzi, V., Mecca, G., & Merialdo, P. (2001, September). Roadrunner: Towards automatic data extraction from large web sites. In *VLDB* (Vol. 1, pp. 109-118).

- Chen, H. H., Tsai, S. C., & Tsai, J. H. (2000, July). Mining tables from large scale HTML texts. In Proceedings of the 18th conference on Computational linguistics-Volume 1 (pp. 166-172). Association for Computational Linguistics.
- Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, 1-38
- Devogele, T., Parent, C. and Spaccapietra, S., 1998. On spatial database integration. *International Journal of Geographical Information Science*, 12(4), pp.335-352.
- Elmagarmid, A.K., Rusinkiewicz, M. and Sheth, A. eds., 1999. Management of heterogeneous and autonomous database systems. Morgan Kaufmann.
- Elmeleegy, H., Madhavan, J., & Halevy, A. (2009). Harvesting relational tables from lists on the web. *Proceedings of the VLDB Endowment*, 2(1), 1078-1089.
- Embley, D., Krishnamoorthy, M., Nagy, G., & Seth, S. (2011). Factoring web tables. *Modern Approaches in Applied Intelligence*, 253-263.
- Embley, D. W., Tao, C., & Liddle, S. W. (2005). Automating the extraction of data from HTML tables with unknown structure. *Data & Knowledge Engineering*, 54(1), 3-28.
- Gatterbauer, W., Bohunsky, P., Herzog, M., Krüpl, B., & Pollak, B. (2007, May). Towards domain-independent information extraction from web tables. In Proceedings of the 16th international conference on World Wide Web (pp. 71-80). ACM.
- Hajmoosaei, A. and Abdul-Kareem, S., 2008. Web data integration system: Approach and case study. In *Business Information Systems* (pp. 410-423). Springer Berlin Heidelberg.
- Hoonlor, A., Szymanski, B. K., Zaki, M. J., & Thompson, J. (2012). An evolution of computer science research. RPI Technical Report 12-01, Rensselaer Polytechnic Institute, Troy, NY.

- Krishna, S. S., & Dattatraya, J. S. (2015, January). Schema inference and data extraction from templated Web pages. In *Pervasive Computing (ICPC), 2015 International Conference on* (pp. 1-6). IEEE.
- Kumar, A. V. (Ed.). (2016). *Web Usage Mining Techniques and Applications Across Industries*. IGI Global.
- Liu, B., Grossman, R., & Zhai, Y. (2003, August). Mining data records in web pages. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 601-606). ACM.
- Lerman, K., Knoblock, C., & Minton, S. (2001, August). Automatic data extraction from lists and tables in web sources. In *IJCAI-2001 Workshop on Adaptive Text Extraction and Mining* (Vol. 98).
- Madhavan, J., Bernstein, P.A. and Rahm, E., 2001, September. Generic schema matching with cupid. In *vldb* (Vol. 1, pp. 49-58).
- M. J. Cafarella, A. Halevy, D. Z. Wang, E. Wu, and Y. Zhang. WebTables: Exploring the power of tables on the web. In *VLDB*, pages 538-549, Auckland, New Zealand, Aug. 2008.
- Nagy, G., Seth, S., Jin, D., Embley, D. W., Machado, S., & Krishnamoorthy, M. (2011, September). Data extraction from web tables: The devil is in the details. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on* (pp. 242-246). IEEE.
- Purnamasari, D., Simri Wicaksana, I.W., Harmanto, S. and Banowosari, L.Y. (2015) 'HTML table wrapper based on table components', *Int. J. Computer Applications in Technology*, Vol. 52, No. 4, pp.237-243.
- Raggett, D., Le Hors, A. and Jacobs, I., 1999. HTML 4.01 Specification. W3C recommendation, 24.
- Rahm, E. and Do, H.H., 2000. Data cleaning: Problems and current approaches. *IEEE Data Eng. Bull.*, 23(4), pp.3-13.

- Rahm, E., & Bernstein, P. A. (2001). A survey of approaches to automatic schema matching. *the VLDB Journal*, 10(4), 334-350.
- Raheja, N., & Katiyar, V. K. (2016). Performance Comparison of Web Data Extraction Techniques Rudy A.G.Gultom,Riri Fitri Sari,Bagio, Proposing the new Algorithm and technique development for Integrating Web Table Extraction and building a Mashup *Journal of Computer Science* 7 (2): 129-142, 2011 ISSN 1549-3636.
- S. Raghavan and H. Garcia-Molina. "Crawling the hidden web," Proc. 27th VLDB Conf., 2001, 129-138.
- Türker, C. (2001). Schema Evolution in SQL-99 and Commercial (Object-) Relational DBMS. In *Database Schema Evolution and Meta-Modeling* (pp. 1-32). Springer Berlin Heidelberg.
- Tuchinda, R., P. Szekely and C.A. Knoblock, 2008. Building Mashup by Example. *Proceeding of the 2008 International Conference on Intelligent User Interfaces, (ICIUI'08)*, ACM, New York, pp: 139-148.
- Wang, J., & Lochovsky, F. H. (2003, May). Data extraction and label assignment for web databases. In *Proceedings of the 12th international conference on World Wide Web* (pp. 187-196). ACM.
- Wang, J., Wen, J.R., Lochovsky, F. and Ma, W.Y., 2004, August. Instance-based schema matching for web databases by domain-specific query probing. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30* (pp. 408-419). VLDB Endowment.
- Yoshida, M., Torisawa, K., & Tsujii, J. I. (2001, September). A method to integrate tables of the world wide web. In *Proceedings of the International Workshop on Web Document Analysis (WDA 2001)* (pp. 31-34).
- Zhai, Y., & Liu, B. (2005, May). Web data extraction based on partial tree alignment. In *Proceedings of the 14th international conference on World Wide Web* (pp. 76-85). ACM.