applied sciences

MDPI

Article

A Novel Approach to Data Extraction on Hyperlinked Webpages

Kamran Shaukat [1,2,3,*], Nayyer Masood [4] and Matloob Khushi [5]

[1] Department of Electrical Engineering and Computer Science, University of Stavanger, 4036 Stavanger, Norway
[2] School of Electrical Engineering and Computing, The University of Newcastle, Callaghan 2308, Australia
[3] Punjab University College of Information Technology, University of the Punjab, Lahore 54590, Pakistan
[4] Department of Computer Science, Capital University of Science and Technology, Islamabad 45750, Pakistan; nayyer@cust.edu.pk
[5] School of Computer Science, The University of Sydney, Sydney 2006, Australia; mkhushi@uni.sydney.edu.au
* Correspondence: Kamran@pujc.edu.pk; Tel.: +47-9864-8407

Received: 22 July 2019; Accepted: 7 November 2019; Published: 25 November 2019

Abstract: The World Wide Web has an enormous amount of useful data presented as HTML tables. These tables are often linked to other web pages, providing further detailed information to certain attribute values. Extracting schema of such relational tables is a challenge due to the non-existence of a standard format and a lack of published algorithms. We downloaded 15,000 web pages using our in-house developed web-crawler, from various web sites. Tables from the HTML code were extracted and table rows were labeled with appropriate class labels. Conditional random fields (CRF) were used for the classification of table rows, and a nondeterministic finite automaton (NFA) algorithm was designed to identify simple, complex, hyperlinked, or non-linked tables. A simple schema for non-linked tables was extracted and for the linked-tables, relational schema in the form of primary and foreign-keys (PK and FK) were developed. Child tables were concatenated with the parent table's attribute value (PK), serving as foreign keys (FKs). Resultantly, these tables could assist with performing better and stronger queries using the join operation. A manual checking of the linked web table results revealed a 99% precision and 68% recall values. Our 15,000-strong downloadable corpus and a novel algorithm will provide the basis for further research in this field.

Keywords: conditional random fields; linked web tables; relational databases; schema extraction; web tables

1. Introduction

Over the years, the World Wide Web (WWW) has gained significant popularity and is presently reckoned to be a treasure trove of information. The plethora of this information takes the form of images, text, audios, videos, etc. The major portion of this information is freely available which benefits the users ingress to the required information coherently. A user poses a search query on a web-browser and acquires a list of possibly related uniform resource locators (URLs) against keyword(s). A tabular representation of data/information on the web is considered more effective and precise than non-tabular. This representation is a two-dimensional representation of facts. Generally, there are two types of web tables, (i) simple and (ii) complex. The simple structure tables contain a header row followed by one or more data rows, whereas complexly structured tables can contain a title, group header, data, and header rows. The number of available tables on the web ranges from hundreds of thousands to millions [1]. This data in the form of tables is brief most of the time, yet is very rich in information. Tables are designed for knowledge management, information retrieval, web mining, summarization, and content delivery to mobile devices [2,3].

The pivot of existing state-of-the-art schema extraction techniques is the transformation of web tables into a relational schema to efficiently retrieve the relevant information against the search query. The contemporary state-of-the-art schema extraction techniques for tabular data played a vital role in enhancing the schema extraction process. These techniques support the conversion of web tables and spreadsheets into a relational database, because of their usefulness for querying data by using simple SQL statements. Furthermore, the schema extraction techniques proposed so far only handle the tables with simple structures, leaving millions of tables with complex structures untouched. None of the techniques proposed has contemplated the conversion of linked web tables into combined, relational schema to enable an advanced queries application. The hyper linkage between two tables on two web pages makes them known as a linked web table. The schema of linked web tables can guarantee maximum relevant information retrieval against the query based on the assumption that the base table linked to another table holds further information regarding the base table.

None of the published articles focused on the linkage of two web tables to enable an enhanced query application and development of schema for linked webpages. This paper is an attempt to overcome that deficiency by identifying, processing, and converting linked web tables into relational schema by using primary/foreign key relationships.

Our research was aimed at extending the Cafarella [4–8] approach named WebTable, by considering the complex table structures. Our technique handles the schema extraction of complex table structures that are available in the millions on the web. We have developed our in-house web crawler to download web pages. We have targeted diversified websites to formulate the dataset. The websites included Wikipedia tables, faculty profiles, finance and commerce transactional tables, actor profiles, and different statistical data. Our algorithm further detects the table tags from the HTML code. Conditional random fields (CRF) have been adopted to identify the header row and return a sequence of row labels. The row labels were fed to our designed automaton model to handle the complex structures by identifying the header row with significant accuracy. Moreover, the linkage between two web tables was identified and primary/foreign keys were established. The individual schema of linked web tables was converted into a relational schema to scrutinize the potential of the combined schema in terms of advanced query application (i.e., SQL join operation). We have deposited our datasets and code to Sourceforge for further research, as described in the Methods section.

2. Related Work

In this section, we review contemporary state-of-the-art approaches that somehow link with our technique. The literature is categorized into three main sections; i.e., table detection, schema extraction and matching, and canonical form conversion.

2.1. Table and Header Detection

Tables are often utilized to present information in a structured manner and numerous attempts were made in the literature to recognize and extract the data from tables. Chen et al. [9] proposed a hidden Markov models (HMMs) and stochastic grammar-based method focusing on web table mining from the huge amount of unstructured hypertext documents. They introduced a method for table separation and recognition. They also interpreted and presented tables. The table mining task was performed in modules as shown in Figure 1.

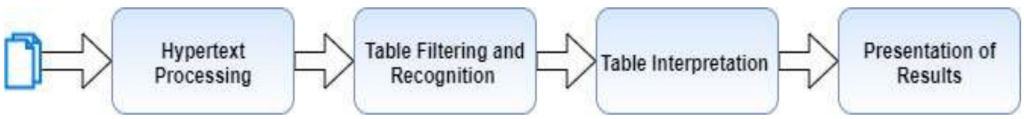

Figure 1. The flow of table mining modified from [8].

There were some limitations in HMMs and stochastic grammar that could negatively affect their performance, whereas CRF were shown to have several advantages over HMMS and stochastic grammar.

Fang et al.'s method [10] focuses on the header detection of different tables available in PDF documents collected from the dataset of CiteSeer [11]. Some techniques focused on table exploring by using table layout characteristics; however, table structure mattered a lot. Approximately, every different format proposed [12,13] had its way of classifying tables available on the web. The header detection algorithm might be able to perform well if different categories of the table could be classified automatically. Sometimes, a table header tag, <th>, contains both a column and row header due to which table header detection procedure is crucial, as shown in Table 1. Multiple levels are also noticeable in a given table header.

Fang [10] used heuristic methods and machine learning techniques for table header detection. This heuristic mechanism utilized the local minimum methods. Their random forest based classifier [14] achieved comparatively better performance than the other heuristic and machine learning techniques. To classify a table as real or non-real, the authors used an efficient and domain-independent machine learning-based table detection algorithm for web documents. Machine learning-based techniques are used to detect real tables, expressed in two-dimensional grid format, i.e., either in rows and columns having logical relations among cells, or in non-real format created for a specific layout for grouping data into condensed clusters as way to show information [15].

Table 1. An example of a complex table header modified from [16].

Form Name	A	B	Total Fields (X = A × B)	Fields with Errors (Y)	Error Rate* (Y/X) × 100 (%)	
					X/Y	% Age
Breast pathology	48	733	35,184	188	0.0053	0.53
Specimen accession	28	667	18,676	63	0.0034	0.34
Patient consent	11	379	4169	11	0.0026	0.26
Demographics	20	348	6960	9	0.0013	0.13
Clinical follow-up	15	1115	16,725	2	0.0001	0.01
Total	122	3242	81,714	273	0.0033	0.33

* The error rate was calculated by expressing the number of fields with errors, as a percentage of the total fields audited for each form type

Table detection, structural analysis, and table interpretation are included by table understanding methods in web documents. Hurst [17] introduced a novel feature set that contained a total of 16 features to analyze the process of web pages. Cafarella [4] identified approximately 150 million HTML tables on the web, mostly containing the structured and related data. Lautert et al. [14] has proposed a primary classification scheme to formalize the structured data (composed of an ordered set of x rows and y columns) into different notations of web tables (WT). Relational knowledge tables were further described in the secondary classification. The authors determined the overall candidate based on the relevant tables to expand the computational approach by presenting a general filtering-based approach they employed for scaling related table computation. Filtering conditions were used in two ways in the algorithm: fewer comparisons and fast comparisons [4,15,18].

The algorithm used tables to represent the data required formally, to get the reader's attention, or for informative comparisons [19]. Th scheme was focused on the basic understanding of the table structure and the identification of its cell components. Statistical relational learning (SRL) was employed to perform the task automatically [20]. To recognize critical cells, Lynx adpted the SRL algorithm [20–22], which is a probabilistic, query-based classifier that uses first-order logic as a representation language. Lynx adopted a selective, featurization approach to feature engineering that kept a feature construction phase with a feature selection process. The experiments were performed on a dataset that consisted of 200 comma separated value (CSV) files, each containing the description of an HTML table selected randomly from 10 large statistical Web sites [23].

2.2. Schema Extraction and Matching

The WebTable [5] project crawled HTML tables, collecting a large amount of web data. The main contribution of [5] was to extract the HTML tables. It is acknowledged that the HTML tables on the web not only contain relational data but are also used for web page layout design. The authors in [6,23,24] provided a survey of schema extraction techniques and continued with follow-up research in that direction. The authors in [8] proposed a relation recovery technique to extract the relations of HTML tables.

Complexly structured tables can contain a title, group header, data rows, and header rows. For example, if a web table contained data about the information about a country's cities, then rows could be grouped by the state or province. Adelfio et al. [25] presented the solution to this problem in the form of an automated method for the schema extraction of tabular data. The classifier was based on the conditional random field [26,27] which was originally used by [28]. The technique they proposed outperformed the WebTable method [8,29]. However, it does not perform the linkage between extracted tables. Tabular data is used in a wide range of applications. For example, Google's main index contains 14.1 billion HTML tables in English documents and the ClueWeb09 dataset contains over 400,000 Excel spreadsheets [8,30]. Sekhavat et al. [31] focused on finding the relationships between the pairs of entities that belonged to the same row.

Many other articles focused on understating the architecture of web tables. The authors in [32] worked on using columns headers to find the relationships between the columns. Venetis et al. [33] proposed a probabilistic model to label the columns and identifying relations between entity columns and the other columns. Chen et al. proposed an extraction system to convert spreadsheet data into relational tuples. Authors in [34] investigated the nature of HTML tables from the web. We have provided the comparison of related techniques in Table 2.

Table 2. Overview of main papers.

Sr#	Paper Title	Published In	Table Detection	Worked On	Corpus Size	Schema Extraction	Linked >Tables
1	Mining tables from large scale HTML	Computational Linguistics, 2000	Yes	HTML	-	No	No
2	Detecting tables in HTML documents	Springer, 2002	Yes	HTML	1393	No	No
3	Uncovering the relational web	Web DB, 2008	Yes	HTML	14.1B	Yes	No
4	Factoring web tables	Springer, 2011	No	Spread Sheets	200	No	No
5	Table header detection and classification	AAAI, 2012	Yes	PDF	200	No	No
6	Schema extraction from tabular data	VLDB, 2013	Yes	HTML, Spread Sheets	7883, 14669	No	No
7	Web table taxonomy and formalization	SIGMOD, 2013	Yes	HTML	30000	No	No
8	Ten years of webtables	Proc. VLDB. 2018	Yes	HTML	-	Yes	No

2.3. Canonical Form Conversion

The indexing property can be used to determine the complex headers in complex tables. Hierarchical categories are extracted by the factorization of isolated headers. A relational database can be built by transforming web tables into a canonical form that could later allow the arbitrary SQL queries to search over induced relational tables. Chen [9] proposed using the <table></table> tag to identify web tables. But most of the time, the <table></table> tag is used to form the page layout. Authors in [21,34] proposed a technique of segmentation by using a technique of minimum indexing point (MIP). Identification of MIP provides four more critical cells (CC1, CC2 for headers, and CC3 and CC4 for data regions respectively) to segment the header and data regions and factors those category headers in rows and columns header, as shown in Figure 2.

	COL1	COL1	COL1	COL1	COL1	COL1
	COL2	COL2	COL3	COL3	COL3	COL3
	COL4	COL4	COL4	COL4	COL5	COL5
	COL8	COL6	COL7	COL8	COL6	COL7
COL9	COL9					
COL9	Col10		$ 100,00			
COL11	COL11					

Figure 2. Example of Indexing by unique header paths.

3. Methodology

This section depicts the detailed methodology employed to acquire the research objectives. All the steps are discussed step by step in the following sections. The following are the methods, their use, and explanations.

(1) Dataset—used for research.
(2) CRF—used for classification of web table rows, and returns a row label sequence.
(3) Automaton—used to classify web tables as relational or non-relational.

3.1. Dataset

15,000 HTML web pages were collected by using our in-house-developed web crawler. This dataset is available at *https://sourceforge.net/projects/linked-web-table/*. Diversified websites were targeted to construct the dataset. The websites included Wikipedia tables, faculty profiles, finance and commerce transactional tables, actor profiles, and different statistical data. Most of the tables on these websites had a web link that directed to another page (possibly) containing another table identifying linked web tables having a schema. The tables in the dataset were either complex or simple structures. The detailed description of the dataset is given in Table 3, rounded to the nearest hundred.

Table 3. Summary of a dataset.

Dataset Characteristics	
Number of Pages	15,000
Number of tables	30,000
Number of pages with tables	12,900
Number of rows	210,000
Number of Complex Tables	18,000
Number of Non-Real Tables	9000
Number of Simple Tables	3000
Number of Real Tables having Links	6000

3.2. HTML Input File

The table headings and data rows were extracted by identifying appropriate combinations of <table><th><td> tags in the HTML files. It was seen that many page layouts were also designed using <table> tags; such designer tables were referred to as non-real tables. It was a challenging task to extract pattern followed by a table tag that possibly contained a relational structure.

Identification of the actual structure from HTML code was also a challenging problem because table/div tags were not necessarily used to depict relational structure. Table/div tags were also used for formatting purposes. Different techniques proposed so far have been given for the direct tables as input for further processing; however, our algorithm extracts the pattern by itself. Non-real tables were identified, described in forthcoming sections, and were skipped. Processing was performed on real tables into two chunks as (i) simple structures or (ii) complex structures. The real table had a two-dimensional grid structure in which logical relation existed among cells.

3.3. Conditional Random Fields

Cafarella [4] proposed a conditional random fields (CRF) based model which was trained on holistic features of a table to classify the web table as either relational on non-relational. The relation-al tables were further classified to obtain whether the first row of the table was a header or not. The WebTable approach excluded the complex structure tables from experiments, which are approximately 32% of them, according to the statistics they provide. The complexly structured tables may contain different types of rows like headers, titles, data rows, or aggregate rows simultaneously. Our technique handles such tables with complex structures to extract the schema and further processing.

Figure 3 is an example of a complex table structure. It contains a row stating high/low risk errors that an item is neither a header row nor a data row. This specific row explains the contents (rows) underneath and the group value of countries listed underneath.

The Error Rate on Different Forms						
Form name	**A**	**B**	**Total fields (X = A × B)**	**Fields with errors (Y)**	**Error rate* (Y/X) × 100 (%)**	
					X/Y	**%age**
High Risk Errors						
Breast pathology	48	733	35,184	188	0.0053	0.53
Clinical follow-up	15	1,115	16,725	2	0.0001	0.01
Patient consent	11	379	4,169	11	0.0026	0.26
Low Risk Errors						
Demographics	20	348	6,960	9	0.0013	0.13
Specimen accession	28	667	18,676	63	0.0034	0.34
Total	122	3,242	81,714	273	0.0033	0.33

Figure 3. Example of a complex table structure modified from [16].

Our technique is an extension of Adelfio [25]. Our technique identifies the types of a rows in a table (i.e., header row, title row, data row, etc.) by following the defined set of rules. After the row type identification, the simple alphabetical structure is assigned to each row, which is depicted in Table 4.

Table 4. Row classes.

Label	Functionality
T	Title of the table mostly table name describing the whole domain of the table.
H	Columns names as cell values define the domain of subsequent data rows beneath.
G	Cluster subsequent rows in a group.
D	Data Tuples/Rows define the header
A	Aggregate/total of above rows.

3.4. Row Class Labels

The rows of simple tables were defined by the H (D) + sequence, whereas the complex structure had variation in patterns, as shown in Table 5.

Table 5. Frequent rows patterns with complex structures.

Sr#	Sequence	Status
1	THD+	Accepted
2	HD+	Accepted
3	H((D) + A)+	Accepted
4	TH((D) + A)+	Accepted
5	H(GD+)+	Accepted
6	TH(GD+)+	Accepted
7	DD *	Rejected
8	DAA *	Rejected

* means zero or many; + means 1 or many

The graphical model of undirected graph CRF was originally introduced by Lafferty, which could classify rows of web tables. We used 24,000 (80%) tables for training; 6000 (20%) tables were used for testing purposes. During testing, the tables were given as input to CRF to label the different row classes. CRF returned the sequence of row labels and fed it to automaton construction. Once the tables were identified as read (relational), they were forwarded further for the schema extraction process.

3.5. Table Schema Extraction

Tables classified as real were processed again to identify the schema; i.e., column names and their data types. In this study, only the following three data types were considered:

1. Varchar: column contains a string data type.
2. Numeric/integer: contains a numeric type of data.
3. Date: contains a date with '/' or '-' separator and data values are marked as the date.

The first row of a table was often identified as a header of a table if <thead> tag was not explicitly mentioned. The data were stored in arrays were and the probability of each row value was checked to be either date or number, but was otherwise classified as varchar.

3.6. Finding the Link Table and Schema

Furthermore, there are several tables which contain hyperlinks on several data values. We are unaware of any technique that focuses on the combined schema of linked web tables. A web link could be found in a header row, at a data value, or the end of a column in a base table. The multiple hyperlink cases are also handled by the approach proposed by applying the defined procedure on the source code of each web page. The identification of linked web tables and their schema generation works according to the following pseudocode Algorithm 1:

Algorithm 1: Table Identification and Schema Extraction Process

```
Input: ← web page
tables[] ← identifying relational tables(web page) // using CRF and Automaton
while(tables[]){
    tables[]← real tables using CRF and automaton
    if(table[] have "a=href")
        fetch_childtable and extract_schema()
        assign PK and FK
    else
        remaining table[] are non-relational
    extract_schema(string table) // Function Definition
        String header ←identify_header(table); //using CRF
        datatype_row ← finding max probability of data type in one column's cell.
        Output: header+datatype_row
        String datatype(string table)
        Cell ← Identifying cells
        if(cell is numeric)
            cell_datatype "numeric"
        else if(cell is date)
            cell_datatype "date"
        else if(cell is varchar)
            cell_datatype "varchar"Function: find_schema(tables);
```

4. Results

On the World Wide Web (WWW), data are often shown in a two-dimensional grid-like structures referred to as tables. HTML web tables can be in different formats, as described by [35]. Schema of such web tables are not explicitly stored as their metadata; hence, it is isolated for later interconnection. Crawlers of search engines cannot access the structure of these web tables and cannot efficiently answer user queries. Our novel method successfully extracted schema for linked tables, as explained in the following sections.

4.1. Identification of Data Tables

We developed a web crawler to collect a comprehensive corpus of 15,000 web pages possibly containing tables. We identified that many web pages used <table> tags to design their layout; therefore, we called such tables non-real tables and the tables with data in them were referred to as real-tables. To identify real tables, the tables in the web pages were extracted and were fed to the CRF classifier. CRF returned a sequence of the row's label as specified in the Methods. These labels were fed to automaton construction (explained in the next section). The automaton output a decision on whether the sequence fed in was a real table or not. Once tables were identified as real tables, they were further processed to extract the schema. The algorithm searched for <a> tags in the row data; if present, the linked web page was read. The linked web page was visited and a linked child-table was identified. Once a linked child table was identified, relational structure schema of both tables was extracted. The parent table's data value was identified as the primary key (PK) and concatenated to the schema of the (linked) child table, the foreign key (FK). Datatypes of the attributes were also identified, as explained in the Methods section.

4.2. Automaton Construction

The row label sequences generated by the CRF model were used as input to the proposed automaton frame-work. The automaton framework NFA was comprised of four main states: start state, intermediate state, dead state, and final state. Figure 4 depicts these connected states as arcs and they are labeled according to the assigned alphabetical string. The start state is highlighted with

yellow, while the intermediate states are highlighted with grey, and final/accepted state has a double border and light green color. The processed input table was declared as a relational table if it reached the final state and was non-relational otherwise.

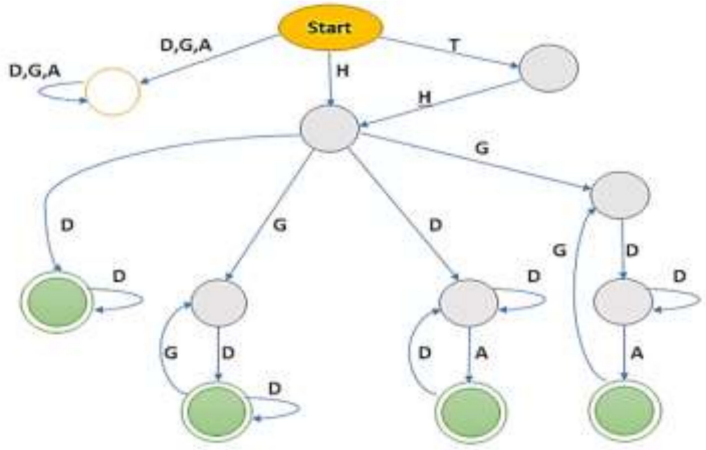

Figure 4. Automaton construction.

The automaton framework followed the two rules described below:

1. If no path existed from one state of a specific row label to reach to another state, then it was non-deterministic, finite-state automaton (NFA), consequently.
2. If all input labels were parsed on the automaton and could not reach a final state, then it could be either on an intermediate state or on the dead state.

4.3. Table Classification

The CRF model was espoused to render labels for the web table rows. As explained above, the CRF classification produced the sequence of labels for candidate table rows. This sequence was parsed to self-defined automaton-NFA which contained frequent possibilities upon which a complex table could be classified as a real table. The automaton-NFA is graphically depicted in Figure 4. We used 24,000 (80%) tables for training and 6000 (20%) tables for testing purposes. Of 21,000 real tables, 14,280 (68%) were predicted correctly (true positive), whereas, of 9000 non-real tables, 8940 (99.3%) were correctly identified. This has been shown in the confusion matrix (Table 6).

Table 6. Confusion matrix.

Actual Class\Predicted Class	Real Tables	Non-Real Tables	Total
Real Tables	14,280 (68%)	6720 (32%)	21,000
Non-Real Tables	60 (0.7%)	8940 (99.3%)	9000
Total	14,340	15,660	30,000

Precision, recall, F-measure, accuracy, and specificity were calculated to scrutinize the potential of the applied framework. The following notation was used:

True positive (TP) = the number of cases correctly identified as real.
False positive (FP) = the number of cases incorrectly identified as real.
True negative (TN) = the number of cases correctly identified as non-real.
False negative (FN) = the number of cases incorrectly identified as non-real

Precision: Exactness: the precision is a fraction of relevant instances among the retrieved instances (1).

$$\text{Precision (P)} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \quad (1)$$

In our case, the precision value reports the number of tuples that the classifier labeled as positive that are actually positive. This measure determines how well the classifier has classified relevant tables only, and does not retrieve any non-relevant items from the dataset. Our precision was calculated at 0.99.

Recall: Completeness: the recall is the fraction of relevant instances that have been retrieved over total relevant instances in (2)

$$\text{Recall (R)} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{2}$$

In our case, the recall value describes how well our classifier does at classifying all the tables in the dataset that have been determined to be topically relevant. The formal representation of precision is presented in Equation (2). Our recall value was 0.68.

Sensitivity: This measure is a true positive recognition rate, which measures the proportion of actual positives that are classified correctly by the classifier. The formal representation of precision is presented in Equation (3). The system achieved a sensitivity of 0.68.

$$\text{Sensitivity (S)} = \frac{\text{TP}}{\text{P}}. \tag{3}$$

Specificity: It reports a true negative recognition rate, which means the proportion of actual negatives that are classified correctly by the classifier. The formal representation of precision is presented in Equation (4). Our specificity was 0.99.

$$\text{Specificity (Sp)} = \frac{\text{TN}}{\text{N}}. \tag{4}$$

Accuracy: The accuracy of a classifier is its ability to differentiate between relevant and non-relevant classes correctly. The system achieved an accuracy of 0.77. The value reports the percentage of test set tuples that are correctly classified by a classifier. The formal representation of precision is presented in Equation (5).

$$\text{Accuracy (A)} = \frac{\text{TP} + \text{TN}}{\text{All}}. \tag{5}$$

F-measure: It is a means of precision and recall. The formal representation of F-measure is presented in Equation (6). The system achieved a F-measure of 0.80

$$\text{F} - \text{measure} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}. \tag{6}$$

4.4. Linked Web Tables

This section illustrates the outcomes of evaluation measures for linked web tables. The precision score for linked web tables was 0.96, which is quite remarkable; however, the recall slightly suffered, as seen with the score of 0.62. We investigated the reason for this lower recall value and discovered that links among tables can be on different places, e.g., a link on a column, link on data value, or link on an aggregate row, whereas the F-measure score, a balance between precision and recall, was 0.74. The sensitivity score was 0.62. The specificity score was 0.96, which we considered very good. A small error rate of 0.24 and an accuracy of 0.75 were achieved.

Table 7 shows the confusion matrix showing the performance of our approach towards finding the linked web tables. The gold standard of manual verification methods was adopted. There was a total of 13,600 links having real tables, of which 2235 were correctly classified. Among 6000 tables in the dataset, 40% that had links, but these links did not contain any real tables. Our classifier had correctly classified 96% of linked tables having only text.

Table 7. Confusion matrix of linked tables.

Actual Class\Predicted Class	Links Having Real Tables	Links Having Simple Text	Total
Links having Real Tables	2235	1365	3600
Links having Simple Text	90	2310	2400
Total	2325	3675	6000

Tables having links on other web pages that contained real tables were classified 62% correctly. An accuracy of 75% was achieved by our classifier as per the dataset; see Table 7. We are not aware of any technique proposed in the literature at present that extracts the schema of linked web tables. Linked web tables that had text on the linked web pages were mostly classified correctly, which enhanced the precision to 96%.

4.5. Comparison with Previous Studies

Figure 5 presents a comparative view of our approach labelled as CRF-A with the results presented by [25]. The classification accuracy for identifying individual row classes (Figure 5A) and real tables (Figure 5B) is shown in comparison with [25]. Our proposed technique outperformed previous studies. Our method obtained an accuracy of 90% for correctly identifying header and data rows, and 85% for the identification of full relational tables. Furthermore, our technique is a step ahead to extracting the combined relational schema that are incomparable. We acknowledge that the datasets used for different studies are not the same due to the unavailability of other research datasets; therefore, we developed our own dataset and made it publicly available for further research. Moreover, previous approaches have not considered linked web tables at all.

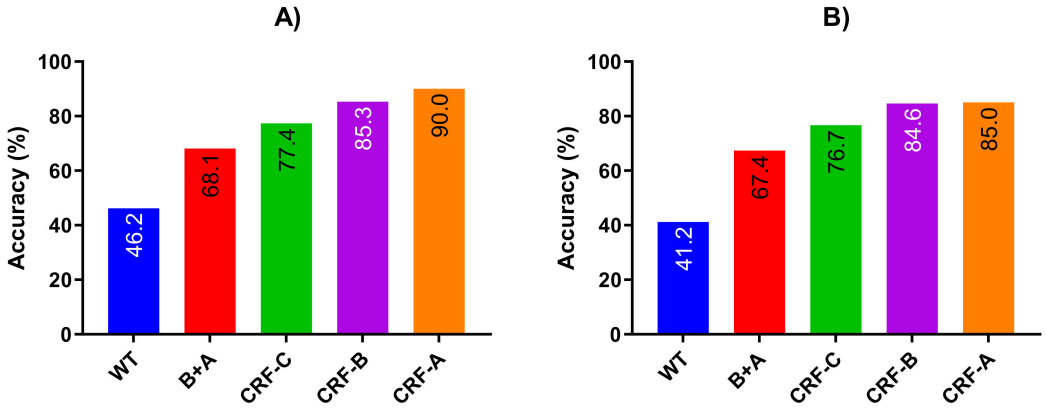

Figure 5. A comparison of five different algorithms. (**A**) Accuracy in the identification of header and data rows. (**B**) Accuracy in the identification of real tables.

5. Conclusions and Future Work

Much of the information on the WWW is presented in unstructured documents [36]. A large amount of relational data exist as tables which are often hyperlinked, providing additional meaning to the data. Lots of work has been done to improve structured extraction [25,37,38] and data cleaning [27]; however, there is still room for improvement in extraction tools. The biggest and most fundamental challenge which has not been fully achieved yet is maintaining the coverage and accuracy of the extraction of a schema for hyperlinked tables. Hence, software architectures need to be enhanced so that they can provide a deeper connection between the users and web tables.

Our schema extraction for data that is hyper-linked on various webpages is an attempt to improve the querying results through search engines. We developed an automatic algorithm that extracts data in web tables, and classifies tables as relational or non-relational, and simple or complex. For relational

tables, it identifies the primary and secondary tables by checking the hyperlink direction. Primary tables are given an identity which is used as a foreign key in the secondary tables. We believe this work will help to improve the effectiveness of queries using SQL and future benchmarking efforts [39]. Relational schema generated for such linked tables should be beneficial for replying to better and more complex queries.

A major problem was faced in extracting the schema of linked tables when links appeared at the beginning or end of the table, where no data row was identified. Though we have achieved very high precision, we only achieved a modest recall which needs further research and work.

Application of our research is not limited to one domain; its application span is wider. If a researcher needs to build any domain specific corpus, then our proposed method can be helpful. Further, it will also enhance the search optimization on search engines such as Google. For example, if we query 'Postal Codes of US States' on Google, the query will return us the information in a tabular form. However, if we query 'Professor with machine learning interest,' instead of returning us the result in a tabular form, the results return us the URLs of profiles of different university faculty members having machine learning as their interest. Professors' information and interests are available on faculty pages that may look like tables but are not in a relational structure. Thus, such information is isolated from further interconnection. Cafarella et al. attempted to handle this but they have only handled simple table structure. We have handled both complex and simple tables.

Therefore, we made our dataset available at https://sourceforge.net/projects/linked-web-table/for other researchers. In the future, we also aim to identify different linking possibilities of tables formed by CSS or dynamic JavaScript and to build a database system by creating the linked relational tables accordingly. This will grant more accessibility to the huge amounts of content on the World Wide Web that are available in the web tables.

Author Contributions: Data analysis and initial draft: K.S.; Methodology: K.S., N.M., and M.K.; First draft preparation, K.S. and M.K.; Writing—Reviewing and editing, K.S., N.M., and M.K.; Supervision: N.M. and M.K.

Funding: This research received no external funding. APS is paid by University of Stavanger, Norway.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Limaye, G.; Sarawagi, S.; Chakrabarti, S. Annotating and searching web tables using entities, types and relationships. *Proc. VLDB Endow.* **2010**, *3*, 1338–1347. [CrossRef]
2. Wang, Y.; Hu, J. Detecting tables in html documents. In Proceedings of the International Workshop on Document Analysis Systems, Princeton, NJ, USA, 19–21 August 2002; pp. 249–260.
3. Zanibbi, R.; Blostein, D.; Cordy, J.R. A survey of table recognition. *Doc. Anal. Recognit.* **2004**, *7*, 1–16. [CrossRef]
4. Cafarella, M.J.; Halevy, A.; Wang, D.Z.; Wu, E.; Zhang, Y. Webtables: Exploring the power of tables on the web. *Proc. VLDB Endow.* **2008**, *1*, 538–549. [CrossRef]
5. Cafarella, M.J.; Halevy, A.; Zhang, Y.; Wang, D.Z.; Wu, E. Uncovering the Relational Web. In Proceedings of the 11th International Workshop on Web and Databases (WebDB 2008), Vancouver, BC, Canada, 13 June 2008.
6. Cafarella, M.; Halevy, A.; Lee, H.; Madhavan, J.; Yu, C.; Wang, D.Z.; Wu, E. Ten years of webtables. *Proc. VLDB Endow.* **2018**, *11*, 2140–2149. [CrossRef]
7. Embley, D.W.; Krishnamoorthy, M.; Nagy, G.; Seth, S. Factoring web tables. In Proceedings of the International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Syracuse, NY, USA, 28 June–1 July 2011; pp. 253–263.
8. Chen, H.-H.; Tsai, S.-C.; Tsai, J.-H. Mining tables from large scale HTML texts. In Proceedings of the 18th Conference on Computational Linguistics, Saarbrücken, Germany, 31 July–4 August 2000; Volume 1, pp. 166–172.
9. Chen, Z.; Cafarella, M. Automatic web spreadsheet data extraction. In Proceedings of the 3rd International Workshop on Semantic Search over the Web, Riva del Garda, Italy, 30 August 2013; p. 1.
10. Fang, J.; Mitra, P.; Tang, Z.; Giles, C.L. Table header detection and classification. In Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, Toronto, ON, Canada, 22–26 July 2012.

11. Caragea, C.; Wu, J.; Ciobanu, A.; Williams, K.; Fernández-Ramírez, J.; Chen, H.H. Citeseer ^{x}: A scholarly big dataset. In Proceedings of the European Conference on Information Retrieval, Amsterdam, The Netherlands, 13–16 April 2014; pp. 311–322.
12. Penn, G.; Hu, J.; Luo, H.; McDonald, R. Flexible web document analysis for delivery to narrow-bandwidth devices. In Proceedings of the Sixth International Conference on Document Analysis and Recognition, Seattle, WA, USA, 13 September 2001; pp. 1074–1078.
13. Han, J.; Pei, J.; Kamber, M. *Data Mining: Concepts and Techniques*; Elsevier: Amsterdam, The Netherlands, 2011.
14. Lautert, L.R.; Scheidt, M.M.; Dorneles, C.F. Web table taxonomy and formalization. *ACM SIGMOD Rec.* **2013**, *42*, 28–33. [CrossRef]
15. Nagy, G. Learning the characteristics of critical cells from web tables. In Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), Tsukuba, Japan, 11–15 November 2012; pp. 1554–1557.
16. Khushi, M.; Carpenter, J.E.; Balleine, R.L.; Clarke, C.L. Development of a data entry auditing protocol and quality assurance for a tissue bank database. *Cell Tissue Bank.* **2012**, *13*, 9–13. [CrossRef] [PubMed]
17. Hurst, M. Layout and language: Challenges for table understanding on the web. Available online: http://wda2001.csc.liv.ac.uk//Papers/12_hurst_wda2001 (accessed on 31 October 2019).
18. Nagy, G.; Padmanabhan, R.; Jandhyala, R.; Silversmith, W.; Krishnamoorthy, M. Table metadata: Headers, augmentations and aggregates. Available online: https://www.ecse.rpi.edu/~{}nagy/PDF_chrono/2010_Padmanabhan_Nagy_etal_DAS2010 (accessed on 31 October 2019).
19. Yakout, M.; Ganjam, K.; Chakrabarti, K.; Chaudhuri, S. Infogather: Entity augmentation and attribute discovery by holistic matching with web tables. In Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, New York, NY, USA, 20–24 May 2012; pp. 97–108.
20. Di Mauro, N.; Basile, T.M.; Ferilli, S.; Esposito, F. Optimizing probabilistic models for relational sequence learning. In Proceedings of the International Symposium on Methodologies for Intelligent Systems, Warsaw, Poland, 28–30 June 2011; pp. 240–249.
21. Esposito, F.; di Mauro, N.; Basile, T.; Ferilli, S. Multi-dimensional relational sequence mining. *Fundam. Inform.* **2008**, *89*, 23–43.
22. Koller, D.; Friedam, N; Džeroski, S; Sutton, C.; McCallum, A.; Pfeffer, A.; Neville, J. *Introduction to Statistical Relational Learning*; MIT Press: Cambridge, MA, USA, 2007.
23. Shaukat, K.; Masood, N.; Mehreen, S. *Population of Data in Extracted Web Table Schema*; LAP Lambert Academic Publishing: Saarbrücken, Germany, 2017.
24. Shaukat, K.; Masood, N.; Mehreen, S.; Haider, F.; Bakar, A.; Shaukat, U. Population of data in web-tables schema. In Proceedings of the 2016 19th International Multi-Topic Conference (INMIC), Islamabad, Pakistan, 5–6 December 2016; pp. 1–6.
25. Adelfio, M.D.; Samet, H. Schema extraction for tabular data on the web. *Proc. VLDB Endow.* **2013**, *6*, 421–432. [CrossRef]
26. Babu, S.; Motwani, R.; Munagala, K.; Nishizawa, I.; Widom, J. Adaptive ordering of pipelined stream filters. In Proceedings of the 2004 ACM SIGMOD international conference on Management of data, Paris, France, 13–18 June 2004; pp. 407–418.
27. Lafferty, J.; McCallum, A.; Pereira, F.C. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. 2001. Available online: https://dl.acm.org/citation.cfm?id=655813 (accessed on 31 October 2019).
28. Condon, A.; Deshpande, A.; Hellerstein, L.; Wu, N. Flow algorithms for two pipelined filter ordering problems. In Proceedings of the Twenty-Fifth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, Chicago, IL, USA, 26–28 June 2006; pp. 193–202.
29. Kodialam, M.S. The throughput of sequential testing. In Proceedings of the International Conference on Integer Programming and Combinatorial Optimization, Utrecht, The Netherlands, 13–15 June 2001; pp. 280–292.
30. Srivastava, U.; Munagala, K.; Widom, J.; Motwani, R. Query optimization over web services. In Proceedings of the 32nd international conference on Very large data bases, Seoul, Korea, 12–15 September 2006; pp. 355–366.
31. Sekhavat, Y.A.; di Paolo, F.; Barbosa, D.; Merialdo, P. Knowledge Base Augmentation using Tabular Data. In Proceedings of the LDOW, Seoul, Korea, 8 April 2014.

32. DiFranzo, D.; Ding, L.; Graves, A.; Michaelis, J.R.; Li, X.; McGuinness, D.L.; Hendler, J. Data-gov wiki: Towards linking government data. In Proceedings of the 2010 AAAI Spring Symposium Series, Palo Alto, CA, USA, 22–24 March 2010.
33. Venetis, P.; Halevy, A.; Madhavan, J.; Paşca, M.; Shen, W.; Wu, F.; Wu, C. Recovering semantics of tables on the web. *Proc. VLDB Endow.* **2011**, *4*, 528–553. [CrossRef]
34. Embley, D.W.; Seth, S.; Nagy, G. Transforming web tables to a RELATIONAL database. In Proceedings of the 2014 22nd International Conference on Pattern Recognition, Stockholm, Sweden, 24–28 August 2014; pp. 2781–2786.
35. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
36. Khushi, M.; Carpenter, J.E.; Balleine, R.L.; Clarke, C.L. Electronic biorepository application system: Web-based software to manage receipt, peer review, and approval of researcher applications to a biobank. *Biopreserv. Biobank.* **2012**, *10*, 37–44. [CrossRef] [PubMed]
37. Hassan, M.U.; Shaukat, K.; Niu, D.; Mahreen, S.; Ma, Y.; Haider, F.; Zhao, X. An Overview of Schema Extraction and Matching Techniques. In Proceedings of the 2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), Xi'an, China, 25–27 May 2018; pp. 1290–1294.
38. Cafarella, M.J.; Halevy, A.; Khoussainova, N. Data integration for the relational web. *Proc. VLDB Endow.* **2009**, *2*, 1090–1101. [CrossRef]
39. Khushi, M. Benchmarking Database Performance for Genomic Data. *J. Cell. Biochem.* **2018**, *6*, 877–883. [CrossRef] [PubMed]