End-to-End Conversion of HTML Tables for Populating a Relational Database

George Nagy
Rensselaer Polytechnic Institute
Troy, NY, USA

David W. Embley
Brigham Young University
Provo, UT, USA

Sharad Seth
University of Nebraska, Lincoln
Lincoln, NE, USA

Abstract— **Automating the conversion of human-readable HTML tables into machine-readable relational tables will enable end-user query processing of the millions of data tables found on the web. Theoretically sound and experimentally successful methods for index-based segmentation, extraction of category hierarchies, and construction of a canonical table suitable for direct input to a relational database are demonstrated on 200 heterogeneous web tables. The methods are scalable: the program generates the 198 Access compatible CSV files in ~0.1s per table (two tables could not be indexed).**

Keywords— table segmentation, table index, Wang category, header factoring, header cross-product, canonical relational table

I. INTRODUCTION

Several successful methods have been demonstrated for locating and delimiting HTML tables in spite of the common use of "table" tags for non-table page layout [1]. For many years spreadsheet software has offered built-in functions to import HTML tables in either a proprietary format that preserves both structure and formatting or as comma-separated values (CSV files) that retain the fundamental grid structure but not most cell formatting.

Table segmentation, i.e., the separation of table headers, notes and footnotes from data cells, has proved more difficult. It typically relies on column-headers located at the top, and row headers at the left of the table, and on only moderately reliable heuristics based on alphabetic header content and numerical data [2,3]. Beyond segmentation, it is even more difficult to assign a unique row and column index (that may span several header rows) to every data cell. Finally, we do not know of any research that attempts to analyze and extract category headers as prescribed by X. Wang [4].

Yet all of the above is necessary in order to populate a database with web table content. The advantage of doing so has long been recognized [5,6]. A DBMS provides query and retrieval functions that allow combining information from several tables in the form of new tables [7]. Although printed and HTML tables are logically symmetric in row and column organization, relational tables are not: rows are *records* (or *tuples*), and columns are f*ields* (or *entities*). This organization opens the way for a wealth of useful operations based on predicate calculus and governed by the well-established concepts of relational algebra and calculus [8]. With the proposed transformation, the uniqueness of table indices is reflected by the notion of *keys*, and the grid structure by the fixed *arity* of tuples.

We demonstrate below three contributions, all of which we consider essential for effective computer search of HTML table contents:

1. Table segmentation based on the fundamental indexing property rather than on appearance features. Our experiments on segmentation of indexing (i.e., minimal) headers and data regions yield 99.5% agreement with human ground truth, as opposed to 85% with appearance features [9]. The single error on 200 tables reflect ground truth judgments and will not hamper database operations.
2. Extraction of category structure based on "factoring" prefixed headers. We detect all 21 multi-category headers in the 200 tables, with no false positives. An example of a multi-category rowheader column is: `[Blois, Male, Female, Tours, Male, Female]`^{tr}. The two mutually exclusive and exhaustive (or orthogonal) categories are `{Blois, Tours}` and `{Male, Female}`. In real tables the categories may be spread over several header rows or columns and their extraction is considerably more complicated.
3. Resolution of the apparent conflict between the symmetric structure of ordinary tables and the asymmetric organization of relational tables by transforming the CSV tables imported from the web into a canonical format suitable for immediate input to a database. The canonical CSV table's rows are assembled by concatenating every row and column header path of the original table with the data cell string that it indexes. This format preserves the category structure. Furthermore it allows, by means of a pivot operation, to use any category for fields (attribute names) and the remaining categories as keys, regardless of the original layout of the table

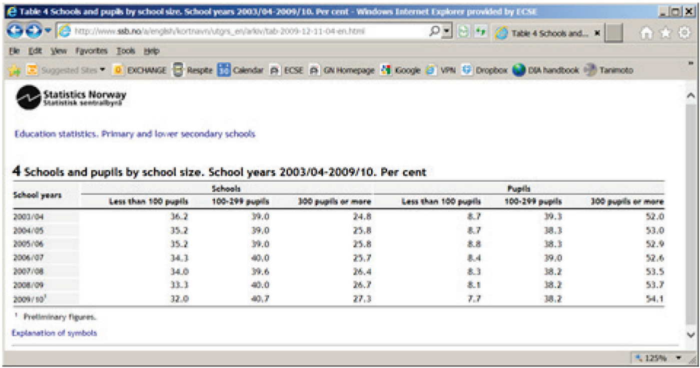

Table 4 Schools and pupils by school size. School years 2003/04-2009/10. Per cent - Windows Internet Explorer provided by ECSE

Statistics Norway
Statistisk sentralbyrå

Education statistics. Primary and lower secondary schools

4 Schools and pupils by school size. School years 2003/04-2009/10. Per cent

School years	Schools			Pupils		
	Less than 100 pupils	100-299 pupils	300 pupils or more	Less than 100 pupils	100-299 pupils	300 pupils or more
2003/04	36.2	39.0	24.8	8.7	39.3	52.0
2004/05	35.2	39.0	25.8	8.7	38.3	53.0
2005/06	35.2	39.0	25.8	8.8	38.3	52.9
2006/07	34.3	40.0	25.7	8.4	39.0	52.6
2007/08	34.0	39.6	26.4	8.3	38.2	53.5
2008/09	33.3	40.0	26.7	8.1	38.2	53.7
2009/10[1]	32.0	40.7	27.3	7.7	38.2	54.1

[1] Preliminary figures.

Explanation of symbols

(a)

	A	B	C	D	E	F	G
1	4 Schools and pupils by school size. School years 2003/04-2009/10. Per cent						
2	School years	Schools			Pupils		
3		Less than 100	100-299 pupils	300 pupils	Less than 100	100-299	300 pupils
4	2003/04	36.2	39	24.8	8.7	39.3	52
5	2004/05	35.2	39	25.8	8.7	38.3	53
6	2005/06	35.2	39	25.8	8.8	38.3	52.9
7	2006/07	34.3	40	25.7	8.4	39	52.6
8	2007/08	34	39.6	26.4	8.3	38.2	53.5
9	2008/09	33.3	40	26.7	8.1	38.2	53.7
10	2009/101	32	40.7	27.3	7.7	38.2	54.1
11	1	Preliminary figures.					
12	Explanation of symbols						

(b)

	A	B	C	D	E	F	G
1	4 Schools and pupils by school size. School years 2003/04-2009/10. Per cent						
2	School years	Schools	Schools	Schools	Pupils	Pupils	Pupils
3		Less than 100	100-299 pupils	300 pupils	Less than 100	100-299	300 pupils
4	2003/04	36.2	39	24.8	8.7	39.3	52
5	2004/05	35.2	39	25.8	8.7	38.3	53
6	2005/06	35.2	39	25.8	8.8	38.3	52.9
7	2006/07	34.3	40	25.7	8.4	39	52.6
8	2007/08	34	39.6	26.4	8.3	38.2	53.5
9	2008/09	33.3	40	26.7	8.1	38.2	53.7
10	2009/101	32	40.7	27.3	7.7	38.2	54.1
11	1	Preliminary figures.					
12	Explanation of symbols						

(c)

Fig. 1. Table segmentation. Row and column headers yellow, CCs green.

Our work leading up to the current process includes examination of egregious tables [10] and studies of table semantics [11,12]. Our 2006 survey [13] is largely obsolete in view of recent work by teams sponsored by Google [14], Yahoo [15], Citeseer [16, 17] and by other academic groups [18, 19, 20]. Current research by Adelfio and Samet on table schema extraction [21] and by Lautert, Scheidt and Dorneles on formalization of the relational aspects and classification of web tables [22] is very much in the spirit of our work. They experiment on much larger data sets than we do, but their analyses are less fine grained.

Because of space limitations, only brief descriptions and examples of our approach to segmentation, category extraction and canonical transformation are given in Sections 2, 3, and 4. In Section 5 we present our experiments and results on tables culled from the web. Section 6 summarizes this research and proposes topics for further investigation.

	A	B	C	D	E
			Less than 100	100-299 pı	300 pupils or n
1	Schools	2003/04	36.2	39	24.8
2		2004/05	35.2	39	25.8
3		2005/06	35.2	39	25.8
4		2006/07	34.3	40	25.7
5		2007/08	34	39.6	26.4
6		2008/09	33.3	40	26.7
7		2009/101	32	40.7	27.3
8	Pupils	2003/04	8.7	39.3	52
9		2004/05	8.7	38.3	53
10		2005/06	8.8	38.3	52.9
11		2006/07	8.4	39	52.6
12		2007/08	8.3	38.2	53.5
12		2008/09	8.1	38.2	53.7
12		2009/101	7.7	38.2	54.1

Fig. 2. An alternative layout for the three categories of the table of Fig. 1.

II. SEGMENTATION

The table is segmented by first finding the Minimum Indexing Point (MIP) [23] and then the four Critical Cells (CC1 and CC2 for the stub header, CC3 and CC4 for the data region) [9]. Cells unmerged to accommodate CSV are left and top filled with the preceding cell content. The algorithm of [23] searches for the MIP (CC2) from the top left corner and backtracks from redundant (for indexing) rows or columns. The remaining CCs are found heuristically by deleting sparse rows from the data region below and to the right of the MIP. The segmentation is exemplified by the small Statistics Norway web table of Fig. 1a. It is shown in Fig. 1b as imported into a spreadsheet in CSV format.

The segmented table with its indexing headers is displayed in Fig. 1c. The MIP (or CC2, here `A3`) is found by searching from cell `A1` for unique column header columns and unique row header rows. The unnecessary rows above the header are eliminated by backward sweep from the MIP of the minimum indexing rows. The CCs in Fig. 1c are `A2`, `A3`, `B4`, and `G10`. Therefore the column header is `B2:G3` and the row header is `A4:A10`.

Rows below CC4 (`G10`) are eliminated. The footnote in `A11:B11` is linked to its reference marker in `A10`. Note that conversion to CSV resulted in demotion of the superscript "1". Empty rows and rows with no data (there are none here) are retained and marked, but not used for segmentation. Prefixing is required when the repetition of cell labels prevents unique indexing. *Prefixing* prepends an additional header row, as required by the column header of Fig. 3.

Blois	Male	Female	Tours	Male	Female

Fig. 3. The repetition of "Male" and "Female" requires prefixing, i.e., the addition of the prefix row [Blois, Blois, Blois, Tours, Tours, Tours] above the prefixed row header.

III. CATEGORY EXTRACTION

Wang categories [4] are fundamental to logical table structure. As explained in Section 4, extracting them increases flexibility in manipulating tables in any relational paradigm. The row category and the two column categories of our example are listed in Fig. 4. These can be extracted by comparing the number of unique elements in the cross-product

RowCat 1	ColCat 1	ColCat 2
2003/04	Schools	Less than 100 pupils
2004/05	Pupils	100-299 pupils
2005/06		300 pupils or more
2006/07		
2007/08		
2008/09		
2009/101		

Fig. 4. Categories extracted from the table of Fig. 1 Only the header depends on the layout: if extracted from Fig. 2, there would be two row categories.

of a pair or header rows with the length of the header. The two rows of the column header of Fig. 1 constitute two categories because the Cartesian product ({Schools, Pupils} × {Less than 100, 100-299 pupils, 300 or more}) of the two rows has 6 elements, which is exactly the length of the column header.

The detection of multiple header categories based on the cross-product has been implemented only for pairs or rows or columns. For greater generality we use header factoring [24], which can handle arbitrary sized headers.

For example, the initial algebraic expression used for "factoring" the column header of the table in Fig. 1, is obtained by tracing the header paths from the left to right:

Sch*<100 + Sch*100-299 + Sch*>300 + Pup*<100 + Pup*100-299 + Pup*>300

Note that the * and + operations in the expression represent vertical and horizontal concatenation, respectively. After factoring, the non-singleton sum terms in the top-level product correspond to the categories:

(Sch + Pup)*(<100 + 100-299 + >300)

IV. CANONICAL RELATIONAL TABLE

The contents of the table of Fig. 1 can be laid out by the table designer in 3x2 different ways, (besides permuting rows and columns), because each of the three categories can form a complete row or column header. Fig. 2 shows an alternative layout for the same table.

It is also possible to lay out the table in with only a single column of data. Part of such a 36-row table is rendered in Fig. 5. Any of the six configurations can be obtained from this M×1 form by *pivoting* [7] one of the *key fields* of the first three columns. This is, therefore, our choice for converting the information into a canonical relational form. The CSV file of Fig. 5 was generated by looping through the rows and columns of the (possibly prefixed) headers, and appending the data values from the original table. The headers of the canonical table, which will become field names in the database, preserve the category structure of the original table. The canonical table is ready for loading into Access or any other database.

Aggregates like TOTAL and Percent Change are position-dependent, therefore it would be best to omit them from any web table before its transformation into canonical form. Any DBMS worth its salt can produce all kinds of aggregates: only the data itself must be preserved. We don't yet have a reliable method for removing aggregates, but others, especially V. Long, have pointed the way [19].

RowCat_1.1	ColCat_1.1	ColCat_2.1	DATA
2003/04	Schools	Less than 100 pupils	36.2
2003/04	Schools	100-299 pupils	39
2003/04	Schools	300 pupils or more	24.8
2003/04	Pupils	Less than 100 pupils	8.7
2003/04	Pupils	100-299 pupils	39.3
2003/04	Pupils	300 pupils or more	52
2004/05	Schools	Less than 100 pupils	35.2
2004/05	Schools	100-299 pupils	39
2004/05	Schools	300 pupils or more	25.8
2004/05	Pupils	Less than 100 pupils	8.7
2004/05	Pupils	100-299 pupils	38.3
2004/05	Pupils	300 pupils or more	53
2005/06	Schools	Less than 100 pupils	35.2
2005/06	Schools	100-299 pupils	39
2005/06	Schools	300 pupils or more	25.8
2005/06	Pupils	Less than 100 pupils	8.8
2005/06	Pupils	100-299 pupils	38.3
2005/06	Pupils	300 pupils or more	52.9
2006/07	Schools	Less than 100 pupils	34.3
2006/07	Schools	100-299 pupils	40

Fig. 5. Part of the canonnical table genererated from the table of Fig. 1.

V. EXPERIMENTS

All of the above operations are part of a Python 2.7 module running under IDLE. The program was used to convert 200 web tables imported earlier into Excel [25,26] to canonical form. Only one segmentation error was detected by comparing the Critical Cells generated with Ground Truth. Whether this particular instance does really constitute an error depends on the acceptability of an empty cell as a column header. The program accepted it, but the Ground Truth did not. (Our earlier attempt at segmentation based on appearance features had a 15% error rate prior to interactive correction [27].) Only 20 two-category headers were found using the cross-product, but all 21 were found by factoring. Neither method generated any false positives.

The tables, all from large international statistical websites, vary greatly in layout, size, and content. The distribution of column-header sizes (after prefixing) is as follows: 69% had only one-row headers; 26.5% had at least one two-row header; and 4.5% (9 tables) had at least one three-row header. Two tables could not be indexed because of repeated rows or columns.

All of the footnotes were found in the 33% of the tables that had them. The program detected 218 reference marks to the footnotes within the body of the tables (some had more than a dozen). It missed them in three tables where the footnote reference marks were not near the end of the cell text.

The entire processing for all 200 tables, including writing the 198 files for input to Access, required 27 seconds. A processed web table is shown in Figs. 6-7. Parts of larger tables are rendered from the imported .CSV files in Figs. 8-9.

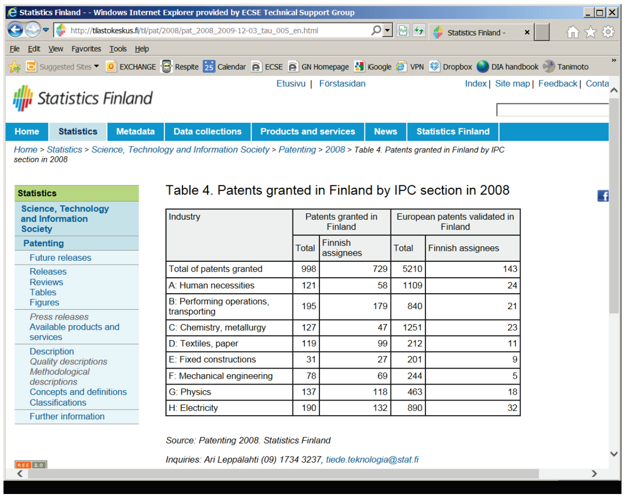

Table 4. Patents granted in Finland by IPC section in 2008

Industry	Patents granted in Finland		European patents validated in Finland	
	Total	Finnish assignees	Total	Finnish assignees
Total of patents granted	998	729	5210	143
A: Human necessities	121	58	1109	24
B: Performing operations, transporting	195	179	840	21
C: Chemistry, metallurgy	127	47	1251	23
D: Textiles, paper	119	99	212	11
E: Fixed constructions	31	27	201	9
F: Mechanical engineering	78	69	244	5
G: Physics	137	118	463	18
H: Electricity	190	132	890	32

Source: Patenting 2008. Statistics Finland

Inquiries: Ari Leppälahti (09) 1734 3237, tiede.teknologia@stat.fi

Fig. 6. A web page from Finland with a common configuration of a two-category column header. The total row contains *aggregates*.

Source: Statistics Canada, 2006 Census of Population.					
Last modified: 2007-10-30					
T51					
Table 4. Patents granted in Finland by IPC section in 2008					
Industry	Patents granted in Fin	European patents validated in Finland			
	Total	Finnish ass	Total	Finnish assignees	
Total of patents granted	998	729	5210	143	
A: Human necessities	121	58	1109	24	
B: Performing operations,	195	179	840	21	
C: Chemistry, metallurgy	127	47	1251	23	
D: Textiles, paper	119	99	212	11	
E: Fixed constructions	31	27	201	9	
F: Mechanical engineering	78	69	244	5	
G: Physics	137	118	463	18	
H: Electricity	190	132	890	32	

Fig. 7. CSV rendition of the three-category HTML table of Fig. 6. Empty cells will be left filled. Same-row prefixing is necessary for the seventh row. The table title is in the fifth row, which is unusual.

Table A-3: Top 10 U.S. Land Ports for Land Trade with Canada and Mexico: 2003 and 2004									
(Thousands of current U.S. dollars)									
Excel | CSV									
U.S. Port	All land modes			Truck			Rail		
	2003	2004	Percent change	2003	2004	Percent chang	2003	2004	Percent ch
U.S.-North American Trade	562,776,436	633,562,711	12.6	392,011,875	452,952,617	15.5	95,724,033	108,360,115	13.2
Top 10 ports	412,424,713	460,654,100	11.7	322,372,568	361,393,164	12.1	79,584,042	87,600,121	10.1
Detroit, MI	101,889,513	113,807,623	11.7	84,810,618	94,019,507	10.9	16,723,319	19,278,278	15.3
Laredo, TX	78,762,959	89,510,852	13.6	54,619,781	63,985,424	17.1	23,940,343	25,398,735	6.1
Buffalo-Niagara, NY	59,369,091	68,351,546	15.1	45,752,599	52,316,608	14.3	9,126,782	10,261,965	12.4
U.S.-Canada Trade	362,319,128	408,612,969	12.8	240,949,027	268,659,618	11.5	64,757,423	74,543,847	15.1
Top 10 ports	288,166,879	323,649,709	12.3	221,837,418	247,417,702	11.5	55,564,511	63,095,059	13.6
Detroit, MI	101,815,113	113,668,714	11.6	84,743,294	93,882,632	10.8	16,718,137	19,276,281	15.3
Buffalo-Niagara, NY	59,275,775	68,283,239	15.2	45,659,600	52,248,579	14.4	9,126,589	10,261,760	12.4
Port Huron, MI	62,244,347	65,879,966	5.8	35,672,586	37,704,369	5.7	22,886,271	23,959,412	4.7
Champlain-Rouses Pt., NY	14,412,634	15,945,026	10.6	12,713,518	14,147,689	11.3	898,156	1,133,615	26.2
Blaine, WA	12,005,376	14,175,533	18.1	9,881,089	11,074,258	12.1	2,098,150	3,092,083	47.4
Alexandria Bay, NY	10,035,184	11,008,768	9.7	10,025,004	11,005,130	9.8	NA	NA	NA
U.S.-Mexico Trade	200,457,309	224,949,742	12.2	163,085,879	184,292,998	13	30,966,610	33,816,269	9.2
Top 10 ports	190,980,524	211,103,066	10.5	158,942,511	179,566,108	13	30,833,262	33,587,526	8.9
Laredo, TX	78,762,959	89,510,852	13.6	54,619,781	63,985,424	17.1	23,940,343	25,398,735	6.1
El Paso, TX	39,204,331	42,779,555	9.1	35,935,405	39,531,129	10	2,472,629	2,928,668	18.4
Otay Mesa, CA	19,678,318	22,188,749	12.8	19,660,724	22,171,883	12.8	NA	NA	NA

RowCat_1.1	RowCat_1.2	ColCat_1.1	ColCat_2.1	DATA
U.S.-North American Trade	ditto	All land modes	2003	562,776,436

Fig. 8. Excerpts from a 36-row table that requires a two-column row index because of duplicate ports under each of three trade headings. There is only one row category because the replication is incomplete. But there are two column categories {All land modes, Truck, Rail} and {2003, 2004, Percemt change}. The canonical table produed has (36 × 9) + 1 = 325 rows. The first two rows are are shown below the table.:

MI	NewPage Corporation	10208	Escanaba Paper Company	Delta	81	103
MI	S D Warren Co	50438	S D Warren Muskegon	Muskegon	51	51
MI	TES Filer City Station LP	50835	TES Filer City Station	Manistee	70	70
MN	Minnesota Power Inc	10686	Rapids Energy Center	Itasca	27	28
MN	Minnesota Power Inc	1897	M L Hibbard	St Louis	73	123
MO	University of Missouri-Columba	50969	University of Missouri Columbia	Boone	6	91
MS	Weyerhaeuser Co	50184	Weyerhaeuser Columbus MS	Lowndes	123	123
NC	Carlyle/Riverstone Renewable Energy	10381	Coastal Carolina Clean Power	Duplin	44	44

Fig. 9. Excerpt from a table where the row header requires three columns because of duplicate entries. `

VI. CONCLUSION

Algorithmic, rather than heuristic, methods were demonstrated for entering CSV tables imported from the web into a relational DBMS. The experiments show that the proposed approach can handle large, complex and heterogeneous tables, from diverse sources, fast enough for production operation. To the best of our knowledge, this is the first end-to-end method for converting arbitrary web tables into an Access-compatible format that preserves the data cell indexing and category structure defined by the headers.

The only heuristics that are part of the program are those used to detect non-data rows above and below the data region. It is possible that these will need modification for new configurations of web tables. The current method depends on the detection of some empty cells or uniform rows (e.g. units) below the header or at the bottom of the table. Fortunately, enlarging the data regions to include these cells does not prevent conversion of the table into a useful database format.

The last two decades have seen many publication describing the conversion of printed, and even-hand-printed, tables to their underlying grid structure. Some of the proposed methods include heuristics for detecting at least column headers. The method suggested here could be applied to imperfectly OCR'd tables provided that the exact cell-content string-matching used in our indexing and category detection schemes is modified to use approximate string-matching. The number of induced indexing errors would depend, of course, on the accuracy of the OCR'd cell contents.

We intend to experiment next with web tables imported into Access to discover what type of useful new information can be queried by combining tables from the same or different sources. We would like to combine factoring and cross-products to spot aggregates, which often give rise to multi-category headers. Another goal is to discover how to make full use of the header hierarchies and auxiliary information (like footnotes and footnote references) that we can already extract. The exploitation of auxiliary items is also an issue in documents without tables.

ACKNOWLEDGMENT

Professor M.K. Krishnamoorthy (RPI) made significant contributions to our analysis of tables and checking of results.

REFERENCES

1 Y. Wang, and J. Hu, Detecting Tables in HTML Documents, Proceedings of the International Workshop on Document Analysis Systems (DAS'02), 249–260, 2002.

2 C. Peterman, C.H. Chang, H. Alam, A system for table understanding, Proceedings of the Symposium on Document Image Understanding Technology (SDIUT'97), 55–62. Annapolis, MD 1997.

3 N. Di Mauro, F. Esposito, and S. Ferilli, Finding Critical Cells in Web Tables with SRL: Trying to Uncover the Devil's Tease, Procs. Int'l Conf. Document Analysis and Recognition (ICDAR'13), Washington, DC 2013.

4 X. Wang, Tabular Abstraction, Editing, and Formatting, Doctoral Dissertation, University of Waterloo, Canada 1996.

5 T. Kieninger, A. Dengel, A paper-to-HTML table converting system, Proceedings of the International Workshop on Document Analysis Systems (DAS'98), Nagano, Japan 1998.

6 W. Gatterbauer, P. Bohunsky, M. Herzog, B. Krupl, B. Pollak, Towards Domain-Independent Information Extraction from Web Tables, Proceedings of the 16th International Conference on World Wide Web (WWW'07), 71–80, Banff, Canada 2007.

7 J. J. Adamski and K.T. Finnegan, Microsoft Access 2010, Course Technology, Boston, MA 2011.

8 J.D. Ullman, Principles of Database Systems, Computer Science Press, 1980.

9 G. Nagy, Learning the characteristics of critical cells form web tables, Procs. Int'l Conf. Pattern Recognition (ICPR'12), Tsukuba, Japan 2012.

10 D. Lopresti and G. Nagy, "A Tabular Survey of Table Processing," Graphics Recognition -- Recent Advances , A. K. Chhabra and D. Dori, Eds., Springer Lecture Notes in Computer Science #1941, 93-120, 2000.

11 D.W. Embley, D. Lopresti, and G. Nagy, Notes on Contemporary Table Recognition, Document Analysis Systems VII, 7th International Workshop, Procs. DAS 2006, H. Bunke and A. L. Spitz, Eds., vol. 3872, LNCS, pp. 164-175, Springer, Nelson, New Zealand, 13-15, 2006.

12 S. Lynn and D.W. Embley, Semantically Conceptualizing and Annotating Tables, Proceedings of the Third Asian Semantic Web Conference (ASWC 2008), Bangkok, Thailand, 2–5, 345 –359, 2009.

13 D.W. Embley, M. Hurst, M. Lopresti, G. Nagy, Table processing paradigms: A research survey, J. Doc. Anal. Recognit. 8, 2–3, Springer, 66-86, 2006.

14 P. Venetis et al. Recovering semantics of tables on the web, Proceedings of the VLDB Endowment 4, 9, 528–538, 2011.

15 N. Dalvi, R. Kumar, B. Pang, R. Ramakrishnan, A. Tomkins, P. Bohannon, S. Keerthi, S. Merugu, A Web of Concepts, PODS'09, Providence, RI, ACM, 2009.

16 Y. Liu, K. Bai, P. Mitra, C.L. Giles, TableSeer: Automatic Table metadata Extraction and Searching in Digital Libraries, Procs. ICDI, Vancouver, Canada, 2007.

17 J.P. Fang, J., P. Mitra, Z. Tang, and C. L. Giles, Table Header Detection and Classification, AAAI, 599–605, 2012.

18 E.C. Silva, A.M, Jorge, L. Torgo, Design of an end-to-end method to extract information from tables. Int. J. Doc. Anal. Recognit. 8, 2, Springer, 144–171, 2006.

19 V. Long. An agent-based approach to table recognition and interpretation, Mcquarie University PhD disseration, 2010.

20 G. Limaye, S. Sarawagi, S. Chakrabarti, Annotating and searching web tables, using entities, types, and relationships, Procs. VLDB Endowment 3, 1–2, 1338–1347, 2010.

21 M. D. Adelfio and H. Samet, Schema Extraction for Tabular Data on the Web, Proceedings of the VLDB Endowment 6, 6, Riva del Garda, Trento, Italy 2013.

22 L. Lautert, M.M. Scheidt, C.F. Dorneles, Web table taxonomy and formalization, SIGMOD Record 42, 3, 28–33, 2013.

23 S. Seth, G. Nagy, Segmenting Tables via Indexing of Value Cells by Table Headers, Procs. Int'l Conf. Document Analysis and Recognition (ICDAR'13), 2013.

24 D.W. Embley, M. Krishnamoorthy, G. Nagy, S. Seth, Factoring Web tables, Procs. EIA/AIE Conf. (F. Esposito, S. Ferilli, eds.), ACM, 2011.

25 R. C. Jandhyala, G. Nagy, S. Seth, W. Silversmith, M. Krishnamoorthy, R. Padmanabhan, 2009. From tessellations to table interpretation. In L. Dixon et al. (Eds.): Calculemus/MKM 2009, Springer-Verlag, Berlin, vol. 5625 of Lecture Notes in Artificial Intelligence, 422-437, 2009.

26 G. Nagy, S. Seth, D.W. Embley, M. Krishnamoorthy, D. Jin, S. Machado, Data Extraction from Web Tables: the Devil is in the Details, Procs. Int'l Conf. Document Analysis and Recognition (ICDAR'11), Beijing, China 2011.

27 G. Nagy, G., and M. Tamhankar,. Vericlick, An Efficient Tool for Table Format Verification, Proc. Document Analysis and Retreival, (DRR'12)., SPIE/EIT, San Francisco, CA 2012.