Interactive Conversion of Web Tables

Raghav Krishna Padmanabhan[1], Ramana Chakradhar Jandhyala[1],
Mukkai Krishnamoorthy[1], George Nagy[1], Sharad Seth[2], and William Silversmith[1]

[1] ECSE, DocLab, Rensselaer Polytechnic Institute, Troy, NY USA 12180
nagy@ecse.rpi.edu
[2] CSE, University of Nebraska-Lincoln, Lincoln, NE USA 68502
seth@cse.unl.edu

Abstract. Two hundred web tables from ten sites were imported into Excel. The tables were edited as needed, then converted into layout independent Wang Notation using the Table Abstraction Tool (TAT). The output generated by TAT consists of XML files to be used for constructing narrow-domain ontologies. On an average each table required 104 seconds for editing. Augmentations like aggregates, footnotes, table titles, captions, units and notes were also extracted in an average time of 93 seconds. Every user intervention was logged and audited. The logged interactions were analyzed to determine the relative influence of factors like table size, number of categories and various types of augmentations on the processing time. The analysis suggests which aspects of interactive table processing can be automated in the near term, and how much time such automation would save. The correlation coefficient between predicted and actual processing time was 0.66.

Keywords: Document Understanding, Interactive Table Interpretation, Performance Evaluation, Ontology Construction, Table Abstraction Tool.

1 Introduction

Our objective is to harvest web tables in order to assist our parent project, TANGO, to construct, with as little human intervention as possible, an ontology in the relatively narrow domain of geopolitics [1]. Since web tables can be readily imported into spreadsheet programs like MS-Excel which provide a natural coordinate system for tables, we developed the Table Abstraction Tool (TAT) to convert Excel tables into Wang Notation [2]. TAT was coded in Visual Basic for Applications (VBA) for ease of access to internal Excel formatting variables. If a table exhibits features that cannot be handled by TAT, then the operator uses Excel commands to change the table into a TAT- admissible format. After verifying the validity of the edited table, TAT creates a category notation which preserves the relationship of the header hierarchies to the content cells. TAT also processes augmentations like aggregates and footnotes tagged by the operator. The Augmented Wang Notation (AWN), which contains both the category information and the augmentations, is embedded into an XML file for portability. The edits can be visually verified by highlighting the relationship between designated headers and content cells. This proofing method was developed in an earlier tool, WNT, which imported web tables into MATLAB rather than Excel [3].

J.-M. Ogier, W. Liu, and J. Lladós (Eds.): GREC 2009, LNCS 6020, pp. 25–36, 2010.

Although we have conducted (and reported elsewhere) experiments on partial automation of data extraction from web tables, we believe that interactive processing will be necessary for some tables for the foreseeable future and that the properties of tables that preclude complete automation are worthy of careful study.

Comprehensive reviews of two decades of research on table processing appear in [4, 5]. Algorithms were first developed for specifying cell location in terms of rulings or, in the case of unruled tables, the geometric alignment and typographic similarity of cell content (e.g., [6,7,8,9]). A recent proposal for an end-to-end system divides the task into table detection, segmentation, function analysis, structural analysis and interpretation, but was not implemented and does not define which tables can and cannot be processed [10]. None of the methods that address web tables (e.g. [11]) carry the analysis to the layout-independent multi-category level. Some of the reasons why we do not expect table recognition to be fully automated in the near future were presented at GREC 1999 [12]. Our model of table processing consists of six interrelated tasks:

Task 1. Table Recognition: Detection of tables within a larger document or corpus, and determination of their exact locations and extents. This is not trivial with unruled Web tables [13].

Task 2. Geometric Structure Extraction: Recognition of the geometric grid structure that characterizes all tables and associated text within the table frame from grid coordinates. Most classical table processing research, especially on scanned tables, addressed this task (e.g. [14]).

Task 3. Table Interpretation: Associating content cells with the heading structure and describing their relationship independently of the geometric layout of the table. This step targets the underlying *logical table.* We have recently developed a formalism to link Task 3 with Task 2 [15].

Task 4. Table Understanding: Determining the conceptual relationships (*is-a, part-of, owns, quantifies, describes*) of the table entries to the contents of other tables, databases, or ontologies. This step, which we call *table understanding* [16], calls for external knowledge from either the vicinity of the table or extraneous sources. It is necessary for conflating tabular data from diverse sources.

Task 5. Metadata Extraction: Extracting and encoding table attributes that do not cleanly fit into either the geometric or the logical views but appear within or adjacent to the table. Examples are *table title, caption, aggregates, footnotes*, and *units*. They have been largely ignored in the table processing literature.

Task 6. Adaptation: Recalling and exploiting the errors and interventions recorded in processing earlier tables to modify the automated aspects of processing the current table. The objective is to develop a system that improves with use, i.e., an evolutionary system that decreases the need for human intervention. Some researchers call this task *learning*.

Here, we present an experimental investigation focused on Tasks 3 and 5. In this experiment 200 tables were randomly chosen from ten large web sites and were processed by one operator.

In Section 2, we list the novel aspects of our interactive procedure. In Section 3, we describe an experimental protocol designed to evaluate the various factors that affect interactive table processing. Section 4 presents the analysis of operator interaction time throughout the processing of the 200 web tables. Section 5 summarizes our observations and offers some projections about what aspects of table processing could be automated in the short term.

2 Novel Aspects

Our work differs from earlier work with respect to

1. Focusing on end-to-end processing of tables from large web sites;
2. Making use of commercial software to import web tables into a spreadsheet and using familiar spreadsheet operations to edit the tables as necessary;
3. Facilitating content analysis by extracting the relationship of headers to content cells rather than only the geometric cell structure;
4. Making provisions for augmentations.
5. Timing, logging, and analyzing all operator interactions.

2.1 Excel Tables

Although several algorithms have been published for finding the cell structure of web tables, with the passage of time this has become a non-issue in research. Excel has built-in provisions for parsing the hypertext and allocating its content to cells. For most sites, it is sufficient to *select* the table, *copy* it, and *paste* into a worksheet. Alternatively, after selection one may use the Excel import menu command. This process is not foolproof. Sometimes the contents of a multi-line table cell are distributed over several worksheet cells, or separate table cells are merged into one worksheet cell. Excel also tries to interpret the data, for instance turning hyphenated numerals into a calendar date. Gratuitous data conversions can be prevented by pre-formatting the target worksheet as *text.* Any errors in conversion must be corrected by the operator. These corrections can be interleaved with the edits necessary to render the table admissible for algorithmic processing by TAT. In the experiments reported below, the interaction time is included under *editing*. In spite of the occasional conversion problems, letting Excel do the heavy lifting has allowed us to concentrate on the more subtle issues.

2.2 Wang Notation

Xinxin Wang in her 1996 dissertation [17] proposed an abstract "table" data type where each logical dimension is defined by a category tree of *labeled domains*. Consider the tables of Fig. 1. The data cell containing "5.0" (a *delta cell* in Wang terminology), is specified by a path through each of the three category trees: DEMOGRAPHICS→IMMIGRANT, YEAR→1990, and COUNTRY→CANADA.

POPULATION IN MILLIONS		DEMOGRAPHICS			
		NATIVE		IMMIGRANT	
		YEAR			
		1990	2000	1990	2000
COUNTRY	CANADA	22.7	25.4	5.0	5.6
	USA	221	249.9	27.4	31.5

(a)

		DEMOGRAPHICS	
		NATIVE	IMMIGRANT
COUNTRY	YEAR		
CANADA	1990	22.7	5.0
	2000	25.4	5.6
USA	1990	221.3	27.4
	2000	249.9	31.5

(b)

Fig. 1. (a) A three-category table; (b) another table with the same Wang Notation

There are several conventions for laying out hierarchical table headings. As row and column headers are conceptually similar, geometric symmetry would suggest that the roles of horizontal and vertical orientations are interchangeable in the layout of table headers (Wang Notation does not distinguish them). However, row headings *above* row subheadings are common in English tables, as in Fig. 1b.

		DEMOGRAPHICS	
		NATIVE	IMMIGRANT
COUNTRY	YEAR		
CANADA	1990	22.7	5
	2000	25.4	5.6
USA	1980	221.3	27.4
	2000	249.9	31.5

(a)

		DEMOGRAPHICS	
		NATIVE	IMMIGRANT
COUNTRY	YEAR		
CANADA	1990	22.7	5
	2000	25.4	5.6
USA	2000	221.3	27.4
	2000	249.9	31.5

(b)

Fig. 2. (a) A well-formed table (WFT) with two categories; (b) not a WFT

A table is *well formed* if it can be represented in Wang Notation. A necessary condition for a well formed table (WFT) is that any combination of paths, one through each category tree, must specify a unique delta cell. Equivalently, the cardinality of the Cartesian product of the unique paths through the category trees must be equal to the number of delta cells (which is eight in all of the above tables).The table on top in Fig. 2 is not a 3-D table, because there is no delta cell that can be specified by the path COUNTRY→CANADA,DEMOGRAPHICS→IMMIGRANT, and YEAR→1980. It is, however, a 2-D WFT, with the category trees of Fig. 3. The table below is not a WFT because the paths COUNTRY→USA, DEMOGRAPHICS→IMMIGRANT, and YEAR→2000 lead to either "27.4" or "31.5".

TAT checks whether a table is *TAT-admissible* before it extracts its Wang category notation. Tables are TAT-admissible even if the roots of some category trees are missing. If the header DEMOGRAPHICS were missing in the table of Fig. 1, TAT would simply generate a unique virtual header *VH xxxxxxxxx* (Virtual Header with the *x*'s indicating the date-time at which the header was generated in the format: *mmdd hhmmss*) as the parent of subcategories NATIVE and IMMIGRANT.

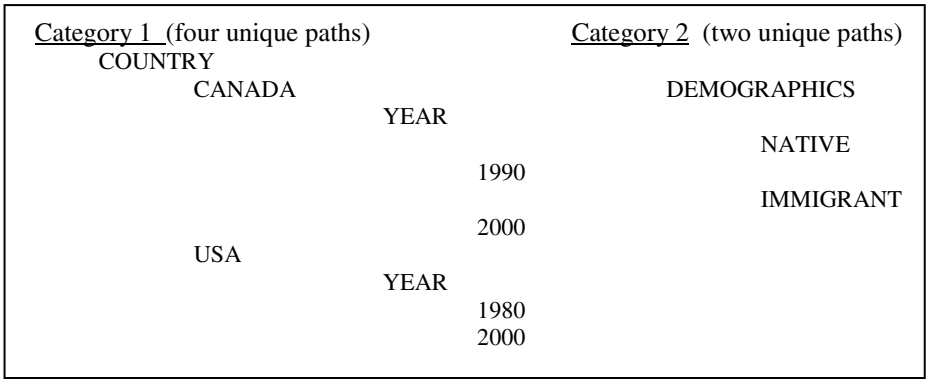

Fig. 3. Category trees of the table of Fig. 2a, shown in TAT's internal indented notation

2.3 Augmentations

An *augmentation* is information appearing in a table that is not part of the header-to-content cell mappings. An augmentation may apply to the entire table (e.g., *Table Title*, *Table Caption, Notes*), to one or more rows or columns (*Unit*), or to a single cell of the table (*Footnote*). The most interesting augmentation is the *aggregate*. For instance, NORTH AMERICA could appear in Fig. 1 instead of COUNTRY (Fig. 4). If no population is listed for NORTH AMERICA, then it is just a header. But if the totals for Canada and USA are listed in that row, then the corresponding paths will be NORTH AMERICA→NORTH AMERICA, NORTH AMERICA→CANADA, NORTH AMERICA→USA. Therefore this aggregate functions both as a category root and as a category leaf cell. Aggregates must be annotated by the operator because TAT cannot yet identify them automatically.

POPULATION IN MILLIONS	DEMOGRAPHICS			
	NATIVE		IMMIGRANT	
	YEAR			
	1990	2000	1990	2000
NORTH AMERICA	244.0	275.3	32.4	37.1
CANADA	22.7	25.4	5.0	5.6
USA	221.3	249.9	27.4	31.5

Fig. 4. The header NORTH AMERICA is both a category root and an aggregate, and must be so tagged in the XML output file

3 Experimental Protocol

We sought to determine the main factors that affect the conversion time of web tables using TAT. We used the collection of rare and unusual tables from Lopresti and Nagy [18] as a guide for selecting tables to evaluate TAT. In particular, we excluded tables that were not well-formed or had any of the following characteristics:

1. Non-rectilinear structure.
2. Text in languages other than English.
3. Cells containing graphic symbols or figures.
4. Recursive structure, i.e., a table with a table as one of its content cells.
5. Concatenation (tables formed by concatenating two or more tables).
6. Sources other than the World Wide Web and formats other than HTML, Microsoft Excel or CSV.
7. Domains other than Geopolitical or Scientific Research data.
8. For convenience, we also excluded tables that span more than one HTML page or Excel sheet.

The experimental protocol was developed in a pilot study. The pilot study was used to determine the final format of the *analysis table,* which would contain all of the experimental data to be collected. A bug in TAT that limited the size of the tables to 100 rows was also found and fixed. The tables used in the pilot study were excluded from the evaluation reported below.

We collected and processed 200 Excel and HTML tables from ten non-profit websites (Table 1). Importing an HTML file into Excel takes negligible time and hence both Excel and HTML tables were treated as one. By "collect", we mean:

1. Browse the websites and save the selected files of tables
2. Store separately the set of 12 tables that we looked at but rejected.
3. Number serially all the accepted tables for subsequent reference and pseudo-randomization.

Table 1. URLs of table sources

Site #	Table Source
1	http://www.statcan.gc.ca/
2	http://www.sciencedirect.com/
3	http://www.worldbank.org/
4	http://www.ssb.no/english/
5	http://www.ojp.usdoj.gov
6	http://www.geohive.com/
7	http://www1.lanic.utexas.edu/la/region/aid/aid98/
8	http://eia.doe.gov/
9	http://ies.ed.gov/
10	http://www.census.gov/population/www/socdemo/voting/cps2006.html

4 Experimental Results and Discussion

The main experiment was conducted in 15 sessions. Processing a table required three consecutive steps:

1. *Editing* the table, i.e., transforming it using Excel operations into TAT-admissible form;
2. *Annotating* the table: this requires clicking on the corner cells of the header and delta cell regions, and on cells containing the table title, caption, aggregates, footnote citations, footnotes, units, and other notes.
3. *Post-processing*, which consists of checking TAT's category assignments by highlighting selected header and delta cells, and either initiating a correction cycle or starting the XML generation algorithm.

Seven tables in the list could not be processed either because they were poorly constructed or because Excel could not interpret their content correctly. Two of these actually failed to match our criteria: they were collected by mistake. One of them was too large (~85000 cells).

The total processing time increases with the number of cells (Table 2) for two reasons: Larger tables typically have more augmentations and require scrolling to edit them into TAT-compatible format. The total processing time includes checking the category assignments by highlighting header and delta cells to display their relationship. It also includes generating the XML output file, which is typically a few seconds.

Table 2. Effect of table size (# of cells in the table) on Total Processing Time

Number of Cells (RxC)	Number of Tables	Avg. Total Processing Time (sec)
< 201	78	134.2
201-400	44	212.3
401-600	25	242.6
601-800	13	430.3
>800	33	394.8
All tables	**193**	**230.5**

On an average, 3-D tables take just 27 seconds more than 2-D tables to process (Table 3). Our data set had too few tables of lower or higher dimensionality for reliable estimates of processing times.

Table 3. Effect of Wang Dimensionality on Total Processing Time

Wang Dimensionality	Number of Tables	Avg. Total Processing Time (sec)
1	2	150.5
2	140	224.3
3	49	251.0
4	2	247.0
All tables	**193**	**230.5**

Table 4 shows that the editing time more than doubles for tables with more than two aggregates compared to tables without aggregates. This does not mean that all the tables with aggregates are not TAT-admissible. But to derive the correct Wang notation/XML, we must transform the table into a form which preserves the category trees (Fig. 4). Tables with aggregates also take much longer to annotate than tables without aggregates. The maximum number of aggregates was 43 in a single table. As illustrated in Fig. 4, aggregates often also serve as top-level row headers. Detecting them requires lexical as well as structural analysis.

Table 5 shows that the presence of footnotes also significantly increases annotation time, but has relatively little effect on editing time. The highest number of footnote cells encountered in a single table was 214. The current implementation requires the user to select each of those cells and annotate it. However, the format of the footnotes below the tables and the corresponding footnote references within the table is uniform enough to allow hope for automated footnote annotation.

The number of cells, the Wang dimensionality, and the prevalence of aggregates and footnotes provide a measure of the amount of operator interaction required to process the table. We predicted the total processing time and the global correlation coefficient by multilinear regression on these four "features." The correlation coefficients between the actual and predicted processing times are shown in Table 6. Sources 6 and 7 contained some poorly constructed/unconventional tables. This resulted in large processing times compared to well-constructed tables with similar features. The global correlation coefficient for all tables without regard to their source was 0.66. The weighted source correlation coefficient was 0.72

Table 4. Effect of aggregates on Editing and Annotation times

Number of Aggregates	Number of Tables	Avg. Editing Time (sec)	Avg. Annotation Time (sec)
0	106	77.8	60.3
1	44	120.2	118.9
2	15	106.5	100.7
>2	28	177.2	171.6
All tables	**193**	**104.1**	**93.0**

Table 5. Effect of footnotes on Editing and Annotation times

Number of Footnotes	Number of Tables	Avg. Editing Time (sec)	Avg. Annotation Time (sec)
0	120	99.9	71.4
1	21	89.8	74.8
2	17	124.3	138.1
>2	35	117.4	155.9
All tables	**193**	**104.1**	**93.0**

Table 6. Source-specific correlation coefficients

Source	Number of Tables	Source-specific correlation coefficient
1	20	0.78
2	15	0.85
3	18	0.97
4	21	0.62
5	24	0.79
6	26	0.40
7	15	0.42
8	24	0.87
9	23	0.72
10	7	0.99
Total	193	0.72

Table 7. Preparation and Action times for each user intervention

Action	Preparation or Idle Time (T_p)	Action Time (T_a)	Ratio (T_p/T_a)
Editing into TAT-admissible form	**6.7**	**97.4**	**0.1**
Annotation	**39.6**	**52.4**	**0.8**
Select Title	3.4	2.5	1.4
Select Caption	0.7	0.9	0.8
Augmentations	5.0	2.6	1.9
Footnotes	3.5	14.6	0.2
Notes	1.8	3.7	0.5
Aggregates	3.2	7.0	0.5
Units	0.7	0.6	1.2
Delta Cell Selection & WFT check	10.4	11.8	0.9
Category Selection	10.9	8.7	1.3
Post-processing	**16.1**	**19.0**	**0.8**
Highlighting (category check)	14.7	11.1	1.3
Generate XML	1.4	7.9	0.2
TOTAL	**62.4**	**168.8**	**0.4**

The time elapsed between the completion of an action and the initiation of the next action was interpreted as the preparation time for the next action. The preparation and action times are shown for each activity in Table 7. As seen earlier, the presence of many aggregates and footnotes significantly increases processing time, and most tables have some of these augmentations. Overall they account for almost 15% of the total processing time, but still 30% less than the fundamental operations of marking category headers and delta cells. Checking the categories assigned by TAT takes significant time (~26 seconds on average). The action time for XML file generation is actually machine time.

5 Summary

TAT was evaluated by a single operator in 15 sessions that took a total of 24.7 hours. The samples were collected from ten web sites which contain thousands of tables relevant to the geopolitical domain. Two hundred tables according to prescribed criteria were processed in a pseudo-random order using TAT. Each selected sample was edited if necessary, and every editing operation was time-stamped and recorded. The time required for editing the table into the desired format along with the interaction to process title, caption, footnotes, units, and aggregates was logged. The Wang Notation for seven of the two hundred tables could not be determined. After processing a table, the operator verified its Wang Notation visually through the TAT functionality which highlights the categories and subcategories associated with the selected delta cell.

Tables with Wang dimensionality 3, which is where layout-independence becomes really significant, took approximately 27 seconds more than tables with Wang dimensionality 2. As expected, there was a strong positive correlation between the processing time and table features (size, dimensionality, aggregates and footnotes). The time distributions have significant positive skew because of a few difficult tables.

Tables with aggregates took much more time than tables without them. Aggregates often result in repeated cells in a header column, which is not TAT-admissible. This requires that the table be modified using Excel commands. The current implementation of selecting and annotating the aggregate cells and footnotes in TAT becomes cumbersome in tables with many aggregates. In the geopolitical domain, it is common to have hundreds of cells with footnote references. Manually selecting these cells is a human intensive, time consuming and error prone task. Thus, there is a great need to automate the identification and annotation of aggregates and footnotes, a task that appears quite feasible. Spanning cells containing *units* should also be relatively easy to detect automatically.

Only a few of the sample tables were processed by TAT without some preliminary editing. The greatest potential savings in time is to make TAT accept a larger variety of table formats. More specifically, it should save the edit sequences applied by operator, generalize them to an arbitrary number of rows and columns, and apply them to new tables in previously seen formats. We are currently working on algorithms to accomplish this [19].

We are also developing methods to automatically determine the delta-cell and header regions, which would save by itself over 15% of the interaction time. We are

exploring the problem of table segmentation using visual cues in the table. The proposed method relies on visual distinctions (typeface, type size, capitalization, alignment) between cells of a table, many of which have been explored in earlier studies by others. The cell's features can be captured in a feature vector with both numerical and categorical attributes. By comparing the feature vectors of adjacent cells using a comparison function, a difference table can be formed and used to perform orientation analysis, category-delta space segmentation and identification of aggregates and footnotes. Automating this phase would pave the way for faster processing and conversion to a layout independent format that would complement the "learning" approach outlined in the previous paragraph.

To determine inter-operator variability in processing time, we are currently planning another experiment with multiple operators on the same corpus of tables. If there is little variability between operators then we can construct an operator-independent formula for predicting processing time as a function of table features.

Conversion of multiple tables from large web sites to Augmented Wang Notation is only the first step towards extracting the intra- and inter-table relationships that are the essential constituents of a domain-specific ontology of semi-structured data.

Acknowledgments. This work was supported by the National Science Foundation under Grants# 041414854 and 0414644 and the Rensselaer Center for Open Source Software. We acknowledge the help of Professor David Embley of BYU in developing the XML formats and the treatment of annotations.

References

1. Tijerino, Y.A., Embley, D.W., Lonsdale, D.W., Ding, Y., Nagy, G.: Toward Ontology Generation from Tables. World Wide Web: Internet and Web Information Systems 8(3), 261–285 (2005)
2. Padmanabhan, R.: Table Abstraction Tool, RPI DocLab, Master's Thesis, May 16 (2009)
3. Jha, P., Nagy, G.: Wang Notation Tool: Layout Independent Representation of Tables. In: Proceedings of the Nineteenth International Conference on Pattern Recognition (ICPR 2008), Tampa (April 2008)
4. Zanibbi, R., Blostein, D., Cordy, J.R.: A survey of table recognition: Models, observations, transformations, and inferences. International Journal of Document Analysis and Recognition 7(1), 1–16 (2004)
5. Lopresti, D., Embley, D.W., Hurst, M., Nagy, G.: Table Processing Paradigms: A Research Survey. International Journal of Document Analysis and Recognition 8(2-3), 66–86 (2006)
6. Sobue, T., Watanabe, T.: Identification of Item Fields in Table-form Documents with/without Line Segments. In: Proceedings of IAPR Workshop on Machine Vision Applications, Tokyo, Japan, November 12-14, pp. 522–525 (1996)
7. Klink, S., Kieninger, T.: Rule-based document structure understanding with a fuzzy combination of layout and textual features. International Journal of Document Analysis and Recognition 4(1), 18–26 (2001)
8. Laurentini, A., Viada, P.: Identifying and understanding tabular material in compound documents. In: Proceedings of the Eleventh International Conference on Pattern Recognition (ICPR 1992), The Hague, pp. 405–409 (1992)

9. Itonori, K.: A table structure recognition based on textblock arrangement and ruled line position. In: Proceedings of the Second International Conference on Document Analysis and Recognition (ICDAR 1993), Tsukuba Science City, Japan, pp. 765–768 (1993)
10. Silva, E.C., Jorge, A.M., Torgo, L.: Design of an end-to-end method to extract information from tables. International Journal of Document Analysis and Recognition 8(2), 144–171 (2006)
11. Krüpl, B., Herzog, M., Gatterbauer, W.: Using visual cues for extraction of tabular data from arbitrary HTML documents. In: Proceedings of the 14th Int'l. Conf. on World Wide Web, pp. 1000–1001 (2005)
12. Lopresti, D., Nagy, G.: Automated Table Processing: An (Opinionated) Survey. In: Proceedings of the Third IAPR International Workshop on Graphics Recognition, Jaipur, India, pp. 109–134 (September 1999)
13. Wang, Y., Hu, J.: Automatic Table Detection in HTML Documents. In: Web Document Analysis: Challenges and Opportunities, October 2003, pp. 135–154 (2003)
14. Handley, J.C.: Table analysis for multiline cell identification. In: Proceedings of Document Recognition and Retrieval VIII (IS\&T/SPIE Electronic Imaging), San Jose, CA, vol. 4307, pp. 44–55 (2001)
15. Jandhyala, R.C., Nagy, G., Seth, S., Silversmith, W., Krishnamoorthy, M., Padmanabhan, R.: From tessellations to table interpretation. In: Carette, J., Dixon, L., Coen, C.S., Watt, S.M. (eds.) Calculemus 2009. LNCS, vol. 5625, pp. 422–437. Springer, Heidelberg (2009)
16. Embley, D.W., Lopresti, D., Nagy, G.: Notes on Contemporary Table Recognition Workshop on Document Analysis Systems. In: Bunke, H., Spitz, A.L. (eds.) DAS 2006. LNCS, vol. 3872, pp. 164–175. Springer, Heidelberg (2006)
17. Wang, X.: Tabular Abstraction, Editing, and Formatting, Ph.D Dissertation, University of Waterloo, Waterloo, ON, Canada (1996)
18. Lopresti, D., Nagy, G.: A Tabular Survey of Automated Table Processing, Graphics Recognition: Recent Advances. In: Chhabra, A.K., Dori, D. (eds.) GREC 1999. LNCS, vol. 1941, pp. 93–120. Springer, Heidelberg (2000)
19. Seth, S., Jandhyala, R., Krishnamoorthy, M., Nagy, G.: Analysis and Taxonomy of Column Header Categories for Web Tables. To appear in Proceedings of the Document Analysis Systems, Boston (June 2010)