Interactive Conversion of Large Web Tables

R. Padmanabhan*, R. C. Jandhyala*, M. Krishnamoorthy*, G. Nagy*, S. Seth+ , W. Silversmith*

** ECSE, DocLab, Rensselaer Polytechnic Institute, Troy, NY USA 12180*
E-mail: nagy@ecse.rpi.edu

+ CSE, University of Nebraska-Lincoln, Lincoln, NE USA 68502
E-mail: 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 recently developed 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 and notes were also extracted. 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 (Wang dimension), 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.

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

1 Introduction

Organizations like the US Census Bureau, the US Department of Agriculture, Statistics Canada, Dow Jones, and even ESPN offer a plethora of well-organized, domain-specific data in tabular form on the web. Except for old data scanned from printed publications, most web tables are presented in HTML, searchable PDF, ASCII, or CSV (spreadsheet) format. The two-dimensional physical layout often represents logically 2-, 3-, or 4-dimensional data, i.e., data where each value is specified by multidimensional category coordinates. Human viewers tend to compare data along one or two of these logical dimensions.

Our objective is to harvest a great many 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 Excel variables. If an Excel table exhibits features that cannot be handled by TAT, then the operator uses Excel commands to change the table into an admissible format. After verifying the validity of the edited table, TAT transforms it into a category notation which preserves the relationship of the header hierarchies to the content cells without regard to the original row-column layout. TAT also processes augmentations like aggregates and footnotes selected by the operator. The Extended Wang Notation (EWN), which contains both the category information and the augmentations, is embedded into an XML file for portability. The visual verification via highlighting of the relationship of the headers to content cells benefited from extensive experimentation with our earlier tool, WNT [3], which was based on Matlab.

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

Task 2. Recognition of the geometric grid structure that characterizes all tables and association of text within the table frame with their grid coordinates. Most classical table processing research, especially on scanned tables, addressed this task e.g. [14]. This step targets the underlying *physical table*.

Task 3. 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 currently call this step *table interpretation*, but the vocabulary of table processing is still fluid. We have recently developed a formalism to link Task 3 with Task 2 [15].

Task 4. 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. 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, units*. As Wang has already noted more than a decade ago, most tables contain one or more of these important attributes but they have been largely ignored in the table processing literature. The integration of such unstructured data with the structured data is the focus of current research in managing comprehensive knowledge bases [17].

Task 6. 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 Task 3 and Task 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 spread sheet 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 Web tables

We conjecture that much of the data used for scholarship or decision making in the geopolitical domain obeys Zipf's Law: a few large sites, each containing thousands of tables, contain the bulk of the data. Most nations publish quantitative data such as the length of rivers, height of mountains, areas of provinces, states, counties, population, age, ethnic origin, birth and death rates, immigration and emigration, education, employment, industrial production, commerce and transportation. We suspect that most of these sites keep and update the data in a conventional database system. We have found that information officers are prepared to discuss only the provenance and interpretation of the data, but not its underlying organization. The middleware and the programmers are hidden behind firewalls.

The majority of these HTML tables are coded consistently in contrast to tables that appear on "amateur" websites such as homepages or blogs. That does not necessarily simplify finding them: the frequent use of the HTML <table> construct for organizing the display of non-table text and graphics has been widely noted [18]. The experiments reported here do not include table detection.

2.2 Excel tables

Although several papers have been published on algorithms 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 preformatting 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 *preprocessing.* In spite of the occasional conversion problems, letting Excel to do the heavy lifting has allowed us to concentrate on the more subtle issues.

2.3 Wang notation

Although Xinxin Wang was not interested in capturing tables but in providing a sound foundation for modifying and rendering tables, her 1996 dissertation influenced much subsequent work [19]. She proposed an abstract "table" data type where each logical dimension is defined by a category tree of *labelled 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, COUNTRY / CANADA. The header hierarchies can be deeper: YEAR could have subcategories JANUARY, FEBRUARY … .

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

		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

Figure 1. Left: a three-category table; Right: 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, geometrical 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. 1, Right.

		DEMOGRAPHICS	
		NATIVE	IMMIGRANT
COUNTRY	YEAR		
CANADA	1990	22.7	5.0
	2000	25.4	5.6
USA	1980	221.3	27.4
	2000	249.9	31.5

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

Figure 2. Left, a 2-D WFT. Right: not an 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 the left of 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, YEAR / 1980. It is, however, a 2-D WFT, with the category trees of Fig. 3. The table on the right is not a WFT because the paths COUNTRY / USA, DEMOGRAPHICS / IMMIGRANT, YEAR / 2000 leads to either "27.4" or "31.5".

Category 1 (four unique paths)

```
COUNTRY
        CANADA
                YEAR
                        1990
                        2000
        USA
                YEAR
                        1980
                        2000
```

Category 2 (two unique paths)

```
DEMOGRAPHICS
        NATIVE
        IMMIGRANT
```

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

A table is *TAT-admissible* if TAT can extract its correct Wang category notation. If a table is TAT-admissible, then it remains admissible if the origin of the grid is shifted or the width of the columns or the height of the rows is changed. More interestingly, it also remains admissible under changes in the depth and branching of the category trees. Tables are also TAT-admissible if the root 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 xxxxxxxxxx* (Virtual Header with the x's indicating the date-time at which the header was generated in the format: *mmdd hhmmss*) as the root of subcategories NATIVE and IMMIGRANT.

2.4 Augmentations

An *augmentation* is information appearing in a table that is not part of the header-to-content cell mappings. An augmentation may refer 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*).

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

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

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 the USA are listed in that row, then the corresponding paths will be NORTH AMERICA / NORTH AMERICA, NORTH AMERICA / CANADA, NORTH AMERICA / USA. Therefore any aggregate functions both as a category root and as a category leaf cell. Aggregates must be annotated by the operator as TAT cannot yet identify any augmentation automatically.

2.5. Transformations of Tables

To check the validity of a table, TAT checks for repeated rows in the *list-row notation* - an array representation of the categories. When there is similar data aggregated over different regions or periods, repeating cells across a column is common. This causes the list-row to contain repeated values. TAT construes this as identical category tree paths - an indication of a malformed table. Thus, to make these tables valid in the TAT sense we transform it using Excel operations (Fig. 5).

National Center for Education Statistics

Table 47-2. Average net access price for full-time, full-year dependent students after grants and loans, by type of institution and family income: 1989–90, 1999–2000, and 2003–04

Type of institution and family income	1989–90	1999–2000	2003–04
Public 2-year			
Total	$7,100	$7,700	$7,700
Low income	5,900	6,100	6,000
Lower middle income	7,500	7,900	7,800
Upper middle income	7,700	8,600	8,700
High income	7,300	8,900	8,800
Public 4-year			
Total	8,700	8,800	9,300
Low income	6,200	5,700	6,000
Lower middle income	8,200	8,200	8,700
Upper middle income	9,300	9,400	10,000
High income	10,500	11,200	11,600
Private not-for-profit 4-year			
Total	14,700	14,000	15,300
Low income	9,100	8,100	10,200
Lower middle income	11,800	11,900	12,400
Upper middle income	14,100	13,400	14,600
High income	20,700	19,700	21,000
Private for-profit less-than-4-year			
Total	10,900	9,600	9,300
Low income	9,500	8,100	8,000
Lower middle income	11,200	10,300	9,700
Upper middle income	12,500	10,700	10,000
High income	14,700	14,000	12,600

National Center for Education Statistics

Table 47-2. Average net access price for full-time, full-year dependent students after grants and loans, by type of institution and family income: 1989–90, 1999–2000, and 2003–04

			1989–90	1999–2000	2003–04
Type of institution and family income	Public 2-year	Total	$7,100	$7,700	$7,700
		Low income	5,900	6,100	6,000
		Lower middle income	7,500	7,900	7,800
		Upper middle income	7,700	8,600	8,700
		High income	7,300	8,900	8,800
	Public 4-year	Total	8,700	8,800	9,300
		Low income	6,200	5,700	6,000
		Lower middle income	8,200	8,200	8,700
		Upper middle income	9,300	9,400	10,000
		High income	10,500	11,200	11,600
	Private not-for-profit 4-year	Total	14,700	14,000	15,300
		Low income	9,100	8,100	10,200
		Lower middle income	11,800	11,900	12,400
		Upper middle income	14,100	13,400	14,600
		High income	20,700	19,700	21,000
	Private for-profit less-than-4-year	Total	10,900	9,600	9,300
		Low income	9,500	8,100	8,000
		Lower middle income	11,200	10,300	9,700
		Upper middle income	12,500	10,700	10,000
		High income	14,700	14,000	12,600

Figure 5. Left, the web table imported into Excel has repeated values in the list-row notation (1-D array) for the row-categories. TAT interprets it as an *invalid* table. Right, the transformed table contains no repeated rows in its row-category list row (20X3 array), preserves the logical structure, and is TAT-admissible.

3. Experimental Protocol

Lopresti & Nagy [20] present a collection of rare and unusual tables which we used as a guide for selecting tables to evaluate TAT. 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 (WWW) and formats other than HTML or Microsoft Excel.
7. Domains other than Geopolitical or Scientific Research data
8. Multiple pages, i.e., tables that span more than one HTML page or Excel sheet.
   This is only for convenience.

In a way, this could be best described as a 'screening experiment' in which the main factors that affect the conversion time of web tables using TAT are sought.

3.1. Table collection:

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:

a) Browse the *Tables* section of the websites and save the files containing the table in a specific location in the file system of the computer that was used to conduct the entire experiment.

b) Store separately the set of tables that we looked at but rejected. There were a total of 12 tables which were rejected. Half of them were rejected because they were concatenated tables and the rest were rejected because they contained graphics/ figures as a part of the delta cells.

c) Make a list of all the acceptable tables and number them serially.

3.2. Table Selection and Processing:

The tables were processed only after all the tables were collected. Processing was carried out in multiple sessions of 1-2 hours each. To avoid any learning from similarities in table construction, the order of processing was randomized. However tables from one source (http://ies.ed.gov/) were all processed consecutively to observe any operator learning. If any of the tables were not TAT-admissible, then they were preprocessed to turn them into TAT-admissible format. Then, header regions, delta cell regions, and augmentations were selected and processed. The XML and log files generated were stored separately.

3.3. Pilot Study:

Since the proposed experiment involved considerable effort and time, we conducted a pilot study to detect any deficiencies present in the method or the tool. The pilot study revealed that TAT could only handle tables with no more than 100 rows. This *bug* in the program was fixed to handle bigger tables. The study also revealed that the operator missed annotating a few augmentations in tables which required much preprocessing. Results of the pilot and detailed descriptions of the processed tables were compiled into an analysis table. The same methodology was used for the actual experiment. The time between the completion

of an action and the initiation of the next action, which is calculated by TAT, was interpreted as the preparation time for the next action to be performed by the user.

Table 1 – URLs of table sources

Site #.	Table Source
1	http://www.statcan.gc.ca/
2	http://www.sciencedirect.com/ [1]
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

[1] The tables from this website belong to the research domain.

4. Experimental Results and Discussion

The main experiment was conducted in 15 sessions. A random number generator was used to order the tables for processing. Seven tables in the list could not be processed either because they were poorly constructed or because their content could not be interpreted by Excel correctly. Two tables did not actually match our criteria; they were collected by mistake and one of them was too large to handle (~85000 cells). Tables 2-5 present the total *processing* time and Tables 6 and 7 present the *preprocessing* (i.e., reformatting) time according to various table characteristics. The *All tables* average in the bottom rows is a simple average over the 193 tables, which corresponds to the weighted average of the values in the same column.

Table 2 - Processing time per table based on Wang dimensionality

Table Feature	Number of tables	Average (sec)	Median (sec)
1-D Lists	2	151	150
2-D tables	140	224	160
3-D tables	49	251	179
4-D tables	2	247	247
All tables	***193***	***231***	***166***

On an average, 3-D tables take just 27 seconds more than 2-D tables to process (Table 2). There were two tables in the collection which were lists in the form of a table (1D tables).

Table 3 presents the effect of aggregates on the processing time of the table. When there are more than two aggregates, the total processing time more than doubles compared to no aggregates. Tables with

aggregates require much more preprocessing than tables without aggregates. Thus, automating the selection and processing of aggregates is an area of research that deserves more attention because it would pave the way for significantly faster table processing.

Table 3 - Processing time per table based on presence of aggregates

Table Feature	Number of tables	Average Time (sec)	Median Time (sec)
Tables with 1 aggregate only	44	271	231
Tables with 2 aggregates only	15	238	188
Tables with > 2 aggregates	28	374	303
Tables with aggregates	87	299	258
Tables without aggregates	106	174	140
All tables	***193***	***231***	***166***
Note: The maximum number of aggregates was 43 in a single table.			

Table 4 –Processing time per table based on presence of footnotes

Table Feature	Number of tables	Average Time (sec)	Median Time (sec)
Tables with 1 footnote only	21	184	179
Tables with 2 footnotes only	17	291	320
Tables with >2 footnotes	35	297	275
Tables with footnotes	73	263	242
Tables without footnotes	120	211	146
All tables	***193***	***231***	***166***
Note: The maximum number of cells with footnotes was 214			

Table 4 shows that the presence of footnotes also affects the processing time. The note in Table 4 indicates that the number of footnote cells could reach as high as 214. The current implementation requires the user to select each of those cells and annotate it. Since there is a specific format for specifying a footnote (i.e., below the delta cells using a footnote reference), automating this process would definitely result in reducing processing time. This also presents an avenue for incorporating *learning* into the system.

Table 5 - Processing time per table based on table size (number of cells in the table)

Table Feature	Number of tables	Average Time (sec)	Median Time (sec)
Tables with number of cells >100 and <= 200	45	148	133
Tables with number of cells >300 and <= 500	36	230	215
Tables with number of cells >500 and <=800	17	384	299
Tables with number of cells >800	33	394	379

Table 5 shows the time taken to process the entire table as a function of table size. The table size criterion presents a strong trend with respect to the total time required to process the table. This is an expected result.

Table 6 - Preprocessing time per table based on presence of aggregates

Table Feature	Number of tables	Average Time (sec)	Median Time (sec)
Tables with 1 aggregate only	43	122	79
Tables with 2 aggregates only	15	106	74
Tables with > 2 aggregates	28	177	126
Tables with aggregates	86	137	103
Tables without aggregates	107	77	60
All tables	***193***	***104***	***75***

Table 6 shows the effect of aggregates on the preprocessing time. There is a more than 100 % increase in the preprocessing time for tables with more than 2 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).

Table 7 - Preprocessing time per table based on table size (number of cells in the table)

Table Feature	Number of tables	Average Time (sec)	Median Time (sec)
Tables with number of cells >100 and <= 200	45	53	47
Tables with number of cells >300 and <= 500	36	115	99
Tables with number of cells >500 and <=800	17	156	118
Tables with number of cells >800	33	186	198

Table 7 shows again the strong positive correlation between the preprocessing time and table size.

5. CONCLUSION

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 that satisfied given 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.

The average total time to process a table was 231 seconds (Median: 166 seconds). The average preprocessing time to edit a table to a desired format was approximately 104 seconds (median: 75 seconds). We understand that a rate of 15 tables per hour per operator is still far too slow for compiling a sizeable ontology, but manual table entry would take at least ten times as long.

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 size. All of the time distributions have a positive skew, indicating that the presence of a few very time-consuming tables.

Processing tables from a single website all at once did not affect the processing time for those tables significantly. In other words, operator learning barely reduced the processing time for tables from the selected source. Tables with aggregates took much more time than tables without them. Aggregates often result in repeated cells in a column, which is not TAT-admissible. This requires that the table be modified to an admissible form 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.

To determine inter-operator variability in processing time, we are currently conducting another experiment with multiple operators on the same corpus of tables. If there is little variability between

operators then we can construct a formula for predicting table processing time as a function of table features. We are also 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.

5 Acknowledgment

This work was supported by the National Science Foundation under Grants# 044114854 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. Y.A. Tijerino and D.W. Embley and D.W. Lonsdale and Y. Ding and G. Nagy, "Toward Ontology Generation from Tables", *World Wide Web: Internet and Web Information Systems*, 8(3): 261 - 285, September 2005.
2. R. Padmanabhan, "Table Abstraction Tool", RPI DocLab, Master's Thesis, May 16, 2009.
3. P. Jha and G. Nagy, "Wang Notation Tool: Layout Independent Representation of Tables", *Proceedings of the Nineteenth International Conference on Pattern Recognition (ICPR'08),*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. D. Lopresti, D.W. Embley, M. Hurst, and G. Nagy, "Table Processing Paradigms: A Research Survey," *International Journal of Document Analysis and Recognition*, 8(2-3): 66-86, Springer, June 2006.
6. T. Sobue, T. Watanabe, "Identification of Item Fields in Table-form Documents with/without Line Segments", *Proceedings of IAPR Workshop on Machine Vision Applications*, Tokyo, Japan, November 12-14: 522-525, 1996.
7. S. Klink, T. Kieninger, "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. A.Laurentini, P.Viada, "Identifying and understanding tabular material in compound documents.", *Proceedings of the Eleventh International Conference on Pattern Recognition (ICPR'92)*, pp.405–409. The Hague, 1992.
9. K.Itonori , "A table structure recongnition based on textblock arrangement and ruled line position.", *Proceedings of the Second International Conference on Document Analysis and Recognition (ICDAR'93)*, pp. 765–768. Tsukuba Science City, Japan, 1993.
10. E.C.Silva, A.M.Jorge, L.Torgo, "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. B. Krüpl, M. Herzog and W. Gatterbauer, "Using visual cues for extraction of tabular data from arbitrary HTML documents", *Proceedings. of the 14th Int'l Conf. on World Wide Web*, 1000-1001, 2005.
12. D. Lopresti, G. Nagy,, "Automated Table Processing: An (Opinionated) Survey", *Proceedings of the Third IAPR International Workshop on Graphics Recognition*, Jaipur, India, 109-134, September 1999.

13. Y. Wang and J. Hu, "Automatic Table Detection in HTML Documents", *Web Document Analysis: Challenges and Opportunities*, pp. 135-154, October 2003.
14. J. C. Handley, "Table analysis for multiline cell identification", *Proceedings of Document Recognition and Retrieval VIII (IS\&T/SPIE Electronic Imaging)*, San Jose, CA 4307: 44-55, 2001
15. R. Jandhyala G. Nagy, S.Seth, W.Silversmith, M.Krishnamoorthy and R. Padmanabhan, "From isothetic tessellations to web tables", *8th International Conference on Mathematical Knowledge Management*, Ontario, Canada, July 2009 (to be published).
16. D. W. Embley, D. Lopresti, and G. Nagy, "Notes on Contemporary Table Recognition Workshop on Document Analysis Systems", *Proceedings of Document Analysis Systems VII, 7th International Workshop*, pp. 164-175, New Zealand, 2006.
17. G. Weikum, G. Kasneci, M. Ramanath, F. Suchanek, "Database and Information-Retrieval Methods for Knowledge Discovery", *Communications of the ACM*, 52(4): 56-64, April 2009.
18. Y. Wang, J.Hu, "Detecting Tables in HTML Documents", *Proceedings of Document Analysis Systems V, 5thInternational Workshop*, pp. 249-260, USA , August, 2002.
19. X. Wang, "Tabular Abstraction, Editing, and Formatting," Ph.D Dissertation, University of Waterloo, Waterloo, ON, Canada, 1996.
20. D. Lopresti, G. Nagy, "A Tabular Survey of Automated Table Processing", *Graphics Recognition: Recent Advances, Springer-Verlag, Berlin, 2000, vol. 1941 of Lecture Notes in Computer Science*, pp. 93-120.