Wang Notation Tool: Layout Independent Representation of Tables

Piyushee Jha, George Nagy
Rensselaer Polytechnic Institute, Troy, NY, 12180, USA
piyushee.jha@gmail.com, nagy@ecse.rpi.edu

Abstract

The Wang Notation Tool (WNT) is a semi-automatic, interactive tool that converts tables to Wang notation – a layout independent representation of tables where all relationships between cells are recorded without relying on the physical structure of tables. WNT requires minimal interaction to select categories from which it deduces relationships. However, if WNT is incorrect, user correction is available to generate correct Wang notation.

1. Introduction

The Semantic Web [1] combines various technologies to supplement the content of web documents with descriptive data (ontologies) that will assist the user in decision making and address their specific needs and wants. The first step of TANGO [2] is to convert a physical table to a layout independent (abstract) form that can be used to create ontologies. This paper describes the Wang Notation Tool (WNT), a semi-automatic tool to convert tables from HTML pages to layout independent Wang notation [3].

An abstract table is specified by an ordered pair (C,δ) where C is a finite set of labeled domains and δ is a mapping from C to the universe of possible values [3]. Therefore, a table has two types of cells: category cells (headers, sub headers) and delta (content) cells, represented by Wang category and Wang delta notation.

WNT requires an understanding of tables and categories. A category consists of a set of cells represented, in abstract form, by a tree. A necessary criterion for tables is selecting categories such that combinations of *any* path from *every* category tree will lead to exactly one delta cell. Some categories don't have a root, making their trees "rootless". In this case, it is necessary to add a *virtual header*, or root. A well-constructed table is one where category and delta cells can be differentiated without any lexical information; if the words were foreign, a user should still understand the relationships in the table based on the structure of cells in the table.

In the next section, we will discuss commonly used words in table processing. Section 3 will detail the interactive system. Section 4 will describe the testing and results. Section 5 will offer some concluding remarks and ideas for future work.

2. Related Work

Some commonly used words in table processing are *detection*, *extraction* [4-5], *interpretation* [5], and *understanding* [7]. There is no universally accepted definition for these words, so we will define them here.

Detecting tables means finding the location and size of the tables in a scanned image or ASCII file. Detection requires layout analysis to find the grid structure that is common to tables. To find the locations of cells within a table, rulings and white spaces are used.

Extraction separates and stores tables from the rest of the document/image. Extraction can mean separating and storing just the table's structure or separating and storing both table structure and table content. The latter requires OCR unless the source is initially electronic. The word *recognition* [8] is also widely used. Recognition consists of both detection and extraction; it is the input needed for interpretation.

Interpretation means obtaining information from a table and presenting that information in a different, sometimes layout independent, way. WNT interprets tables in multiple ways, as trees and Wang notation. Interpretation can also mean creating table models, among which Wang's table model [3] is the most complete.

Understanding tables has not been widely explored. Humans understand tables by connecting information from within a table with all other information they know. Understanding is a difficult task to accomplish with computers; however TANGO attempts understanding. Information from all tables processed within TANGO will be conglomerated into a comprehensive ontology describing the relations within and between each table, thus enabling a computer to "understand" tables.

3. Description of Interactive System

There were many early versions of WNT [9]; each successive version reduced the number of interventions by the user, from over a hundred in the early version to about 10 in the current version for Table 1. WNT detects tables in HTML pages by searching for <table> tags, and then extracting the table into an ASCII file. The ASCII file is sent to a Matlab program which prompts the user to select categories, correct categories, and verify if the relationships within the entire table are correct. A log that records every button click by the user and the time between each step is maintained.

3.1. Category Notation

Generating category notation is the most significant part of WNT because it requires user intervention and the delta notation stems directly from the category notation. The category notation records all the cells within a table that are category cells as well as the relationships between those cells. It is not necessary for category cells to be related lexically; however, they must be related structurally. There are four steps to generate category notation: interactive category construction, intermediate category processing, error correction by user, and category notation generation.

Table 1, from Wang's dissertation [3], is used to demonstrate WNT. Figure 1 is the initial table, with all merged cells split and repeated. The user constructs categories by indicating which cells are category cells and which categories they belong to. To reduce the number of interventions, WNT exploits the fact that all cells belonging to the same category are contained within a rectangle. Therefore, the user clicks on the top-left cell and bottom-right cell corresponding to the rectangle that defines each category (Figure 2). In addition, the user has the option to undo clicks to counter mistakes.

The relationships of the cells within each category must be determined. WNT creates trees and a corresponding *table of contents* for each category based on the location of the cells in the original table relative to each other. Each category tree is displayed as indented notation that users can correct if necessary (Figure 3). The user is free to manipulate the tree describing a category in any way, making it possible to correct almost any error made by WNT. The error correction GUI has the following options: *Undo Last*, *Add Row*, *Add Column*, *Delete Row*, *Delete Column*, *Clear Cell*, *Rename Cell*, *Add Virtual Header*, and *Notation is Correct*.

Wang category notation records the cells in a certain order and then, adds symbols to delineate the relationships between those cells. This ordering is done by converting general category trees to equivalent binary trees and then traversing the trees depth first. Guidelines were developed for the correct placement of delimiters in Wang notation.

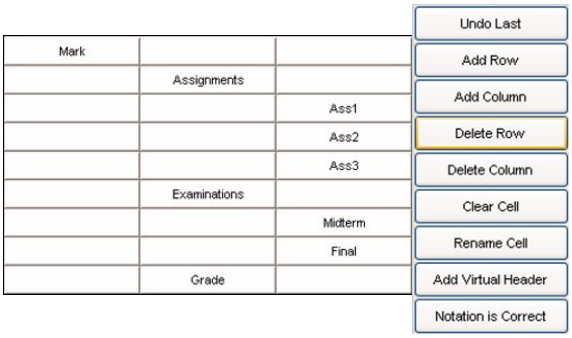

Figure 3: Indented Notation & Error Correction

3.2. Delta Notation & Verification

Delta notation consists of an aggregation of a single path from every category that describes each delta cell. In a well-constructed table, there is exactly one delta cell associated with every possible aggregation of paths through all category trees.

To generate delta notation, a single tree describing the entire table is created, along with a corresponding table of contents. The delta notation is generated by searching for all leaf cells in the same row and column as the delta cell in the original table. Then, the table of contents is searched for those leaf cells and, working backwards, the entire path is determined for each category. Finally, all the paths are assembled with the right symbols to generate Wang delta notation.

Verification is done visually by a GUI that appears after all processing is finished. The user can click on any cell any number of times. All category cells related to a clicked delta cell will change colors to reflect relationships. Alternatively, all delta cells related to a clicked category cell will change colors. If the user is not satisfied, the process may be restarted.

Table 1: Wang Table

Year	Term	Mark					
		Assignments			Examinations		Grade
		Ass1	Ass2	Ass3	Midterm	Final	
1991	Winter	85	80	75	60	75	75
	Spring	80	65	75	60	70	70
	Fall	80	85	75	55	80	75
1992	Winter	85	80	70	70	75	75
	Spring	80	80	70	70	75	75
	Fall	75	70	65	60	80	70

Year	Term	Mark	Mark	Mark	Mark	Mark	Mark
Year	Term	Assignments	Assignments	Assignments	Examinations	Examinations	Grade
Year	Term	Ass1	Ass2	Ass3	Midterm	Final	Grade
1991	Winter	85	80	75	60	75	75
1991	Spring	80	65	75	60	70	70
1991	Fall	80	85	75	55	80	75
1992	Winter	85	80	70	70	75	75
1992	Spring	80	80	70	70	75	75
1992	Fall	75	70	65	60	80	70

Figure 1: Initial table as seen by user

Year	Term	Mark	Mark	Mark	Mark	Mark	Mark
Year	Term	Assignments	Assignments	Assignments	Examinations	Examinations	Grade
Year	Term	Ass1	Ass2	Ass3	Midterm	Final	Grade
1991	Winter	85	80	75	60	75	75
1991	Spring	80	65	75	60	70	70
1991	Fall	80	85	75	55	80	75
1992	Winter	85	80	70	70	75	75
1992	Spring	80	80	70	70	75	75
1992	Fall	75	70	65	60	80	70

Figure 2: Table with categories marked

4. Testing and Results

Preliminary testing on two subjects resulted in cosmetic changes to make WNT user-friendly. Further testing was conducted on six subjects with 17 tables each. All subjects were trained with a Power-Point presentation covering table concepts and a step-by-step example on how to use WNT. Five tables were demonstrated to the subject, after which the subjects processed 17 tables with no input from the author. All complete and partial attempts were logged. Matlab 7.0 or higher is required to run WNT.

The 17 tables [9] tested consisted of 5-6 different kinds of tables; some well-constructed and some badly constructed. Wang notation was generated in 85% of all attempts and was generated *correctly* in 65% of all attempts. The dark gray bars in Figures 4-5 represents the percent of all attempts where Wang notation was generated correctly. The light gray bar represents the percent of all attempts where Wang notation was generated incorrectly.

Wang notation could not be generated when the subject made a large number of corrections, producing invalid trees. Wang notation was generated incorrectly by some subjects who did not understand the concept of virtual headers. The top third of Figure 6 shows part of a badly constructed table. The top category does not have a root. There are two possible solutions, illustrated in the second and third parts of Figure 6. The user can either modify the indented notation to make *Females* the root, with the years as its subcategories, or add a virtual header titled 'Year' on top of the existing table, thus creating a root. Users were confused because the cells within a category do not have to be related lexically.

Subjects also had trouble picking categories correctly. Figure 7 shows two tables that are the same structurally; however, the table on the left has two categories (the first three columns, and Pop.) whereas the table on the right has three categories (State, Year, M/F, and Pop.). Repeated values in the right hand table allow three categories that satisfy the necessary criterion for selecting categories. The gray cells represent delta cells.

The first eight tables were well-constructed and were always generated correctly when generated at all; more challenging tables were often incorrect. User time accounted for 98% of total processing time. The amount of time to process a table was directly related to how well-constructed the table was.

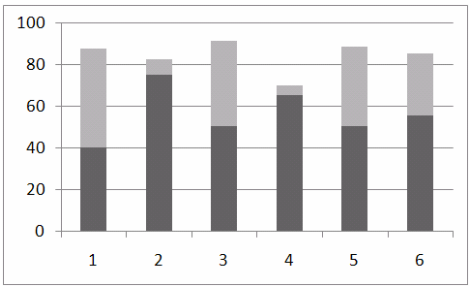

Figure 4: WNT Results by Subject

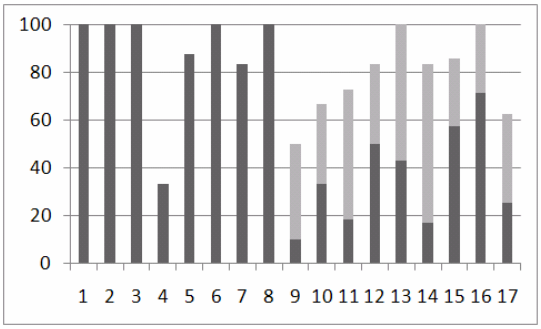

Figure 5: WNT Results by Table

	2000	2001	2002	2003	2004
			Females		
Canada	103,326	105,207	111,027	118,467	124,830
Newfoundland and Labrador	1,773	1,755	1,749	1,830	1,989
Prince Edward Island	345	411	363	429	453

	Females				
	2000	2001	2002	2003	2004
Canada	103,326	105,207	111,027	118,467	124,830
Newfoundland and Labrador	1,773	1,755	1,749	1,830	1,989
Prince Edward Island	345	411	363	429	453

	YEAR (Virtual Header)				
	2000	2001	2002	2003	2004
			Females		
Canada	103,326	105,207	111,027	118,467	124,830
Newfoundland and Labrador	1,773	1,755	1,749	1,830	1,989
Prince Edward Island	345	411	363	429	453

Figure 6: Virtual Headers

STATE	COUNTY	TOWN	POP.
New York	Rensselaer	Troy	
		Brunswick	
	St. Lawrence	Potsdam	
		Canton	
California	San Diego County	Coronado	
		Del Mar	
	Los Angelos County	Malibu	
		Compton	

STATE	YEAR	M/F	POP.
New York	2000	M	
		F	
	2001	M	
		F	
California	2000	M	
		F	
	2001	M	
		F	

Figure 7: Unique Categories

5. Conclusion

WNT is a fast and robust tool for generating Wang notation, especially for experienced users. The users tested were naïve, but upon detailed feedback after the test session, all of them understood how WNT worked. Further training would greatly improve results. Future work could remove the category construction step altogether. This could be done by exploring foreign tables to determine structural patterns that separate category and delta cells. In addition, adaptive learning would benefit the error-correction process. The results from WNT are also converted to an XML document so they can be used by other groups. WNT is now being used for ontology-related applications, such as Query By Table [10] and for the automatic generation of ontologies [11], which is the next step of TANGO, conducted at Brigham Young University.

Acknowledgement: This work was supported by the National Science Foundation under Grant# 044114854.

References

[1] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," *Scientific American*, May 2001.

[2] Y.A. Tijerino, D.W. Embley, D.W. Lonsdale, Y. Ding, and G. Nagy, "Toward Ontology Generation from Tables," *World Wide Web*, 8(3):261–285, Sept. 2005.

[3] X. Wang, "Tabular Abstraction, Editing, and Formatting," Ph.D Dissertation, University of Waterloo, Waterloo, ON, Canada, 1996.

[4] W. Kornfield and J. Wattecamps, "Automatically Locating, Extracting and Analyzing Tabular Data," *Proceedings of 26th ACM SIGIR*, pp. 347-348, Melbourne, Australia, 1998.

[5] A.C. e Silva, A. M. Jorge, and L. Torgo, "Design of an end-to-end method to extract information from tables," *International Journal on Document Analysis and Recognition*, 8(2):144-171, June 2006.

[6] W. Gatterbauer, P. Bohunsky, M. Herzog, B. Krupl, and B. Pollak, "Towards Domain-Independent Information Extraction from Web Tables," *World Wide Web*, pp. 71-80, Banff, Canada, May 2007.

[7] D.W. Embley, M. Hurst, D. Lopresti, and G. Nagy, "Table-Processing Paradigms: a Research Survey," *International Journal on Document Analysis and Recognition*, 8(2):66-86, June 2006.

[8] T. Watanabe, Q. Luo, and N. Sugie, "Layout Recognition of Multi-Kinds of Table-Form Documents," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(4):432-445, April 1995.

[9] P. Jha, "Wang Notation Tool: A Layout Independent Representation of Tables," M.S. Thesis, Rensselaer Polytechnic Institute, Troy, NY, May 2008.

[10] R. Padmanabhan, and G. Nagy, "Query By Table", *submitted to ICPR*, 2008.

[11] S. Lynn, and D.W. Embley, "Automatic Generation of Ontologies from Canonicalized Web Tables", *submitted manuscript*, March 2008, http://tango.byu.edu/.