D.W. Embley, S. Seth, M. Krishnamoorthy, G. Nagy, Clustering header categories extracted from web tables, Procs. SPIE/IST Document Recognition and Retrieval, Feb. 2015.

Clustering header categories extracted from web tables

George Nagy[a*], David W. Embley[b], Mukkai Krishnamoorthy[c], Sharad Seth[d]

[a, c] Rensselaer Polytechnic Institute. Troy, NY, USA
[b] Brigham Young University, Provo, UT, USA
[d] University of Nebraska – Lincoln, Lincoln, NE, USA

ABSTRACT

Revealing related content among heterogeneous web tables is part of our long term objective of formulating queries over multiple sources of information. Two hundred HTML tables from institutional web sites are segmented and each table cell is classified according to the fundamental indexing property of row and column headers. The categories that correspond to the multi-dimensional data cube view of a table are extracted by factoring the (often multi-row/column) headers. To reveal commonalities between tables from diverse sources, the Jaccard distances between pairs of category headers (and also table titles) are computed. We show how about one third of our heterogeneous collection can be clustered into a dozen groups that exhibit table-title and header similarities that can be exploited for queries.

Keywords: Analysis of CSV tables, Segmentation, Component classification, Category structure of table headers, Similarity of table categories and titles, Jaccard distance, Sequential clustering

1. INTRODUCTION

Extraction and analysis of the multidimensional and multi-row/column indexing headers of data from human-readable tables is essential for querying such data. The table headers can always be decomposed into a set of two or more categories that correspond to the orthogonal axes of a data cube. For instance, agricultural production (in euros or dollars or tons) can be considered in terms of countries, years and commodity (wheat, corn). Tables that share identical or similar categories can be combined along the corresponding dimension and are good candidates for participating in the same query. If two tables have the same list of countries and years, but one covers wheat and corn and the other soybeans and peanuts, then it may be possible reach some conclusion about the relationship between grain and oilseed production. On the other hand, commonality of commodities and years but different countries would allow geographic ranking of grain or oilseed production. Queries on tables with the same countries and commodities but different years could lead to temporal insights.

Even though the automatic conversion of web tables and spreadsheets into Comma Separated Variable (CSV) format loses most appearance features (layout and cell formatting), the remaining structural and alphanumeric information is sufficient for subsequent algorithmic analysis. Using the fundamental indexing property of header paths, we segment the CSV tables into nine types of regions, of which the essential ones are the table title, row header, column header, and data regions. We transform web/CSV tables of heterogeneous format and content into uniform data structures that connect the data values with their category labels. The data structures are saved in the form of CSV tables that are imported into Microsoft Access (a relational database management system) and Protegé (open-source software for manipulating RDF (Resource Description Framework) triples and reasoning over OWL (Web Ontology Language) ontologies.

Building on the above work, we now propose an application of a distance measure between pairs of categories from different tables. Our measure is the Jaccard distance i.e., the ratio of the number of shared unique words to the number of unique words in the union of the word lists of the two categories. We apply the same measure to the unique words of each table title, which can therefore be compared to other table titles or to the row or column category headers. We present results from a simple clustering algorithm. Although we retain all data values, we make no use of them for clustering.

Section 2 is a literature review. In Section 3 we give toy examples that demonstrate both our processing steps and the format and content of the input and output files. We must resort to toy examples because even relatively small web tables generate unreadable illustrations. We present our experiments on 200 web tables in Section 4, which also contains illustrations from real data. Section 5 contains a brief summary and some forward looking statements.

* Send correspondence to G.N., nagy@ecse.rpi.edu

2. LITERATURE REVIEW

This literature review has five parts. We provide some pointers to clustering and unsupervised learning. We review X. Wang's pioneering research which has long guided our approach to table understanding. We cite early work on physical table analysis that drew attention to the immense variability in the layout of tables intended for quick human interpretation in contrast to the uniform structure of relational tables. We comment on research that aims, like ours, at higher-level, "logical" analysis of tables. Finally, we summarize our own previous work that underlies our current endeavors.

2.1 Clustering

Computerized clustering (aka unsupervised classification or learning) is often traced back to McQueen [1] and Ball and Hall[2], or to numerical taxonomy, though of course the general notions of taxonomies are rooted in antiquity. Among the early books on the subject were Hartigan's *Clustering Algorithms*[3] and Dubes' and Jain's 1976 classic *Clustering Techniques*[4] (updated in 1999[5]) The topic is presented in every text on pattern recognition or machine learning, though perhaps in none as extensively as in Theodoridis and Koutrumbas[6]. Hundreds of algorithms have been published: graph-theoretic or vector-space based, probabilistic, neural or genetic, flat or hierarchical, agglomerative or divisive, single-link or multi-link, crisp or fuzzy, expectation maximization, kernel or density based. Recent research on content-based document clustering is often based on latent semantic indexing via singular value decomposition[7,8].

Clustering algorithms implicitly or explicitly attempt to optimize some objective function that maximizes the separation of the clusters and minimizes their girth, subject to constraints on the number, population or shape of the clusters. There is seldom any guarantee of attaining the global extremum. Sequential algorithms make one or more passes over the data, comparing each successive pattern to already clustered patterns or to some representative of each existing cluster. The comparison may be based on a metric distance function or on a similarity measure that does not obey the triangle equality.

The simplest and fastest clustering algorithm that we know is a single-pass sequential algorithm[9]. The first pattern is assigned to Cluster #1, and successive patterns, in an arbitrary order, are either assigned to the closest existing cluster if it is near enough, or become the seed of a new cluster if all the existing clusters are too far. The only parameter is the threshold for creating new clusters, but there are many options for measuring the proximity of a pattern to a cluster. A two-threshold variant of the algorithm leaves patterns that fall between the two thresholds unassigned, either permanently or until subsequent passes, with converging thresholds, assign them. A measure of the validity of the generated clusters is that the stability of their membership proved stable under random permutations of the presentation sequence. The singular appeal of this method is its speed: there are millions of tables on the web whose headers are potentially subject to clustering.

2.2 Table Categories

X. Wang formalized the distinction between physical and logical structure in the course of building X-Table for practical table composition in a Unix X-Windows environment[10]. She defined *layout structure* as the presentation form of a table, and *logical structure* as a set of labels and entries. The *logical structure* of a table was formulated by Wang in terms of category trees corresponding to the header structure of the table. "Wang categories", a form of multidimensional indexing, are defined implicitly by the 2-D geometric indexing of the data cells by row and column headers. The index of each data cell is unique (but it may be multidimensional and hierarchical in spite of the flat, two-dimensional physical layout of the table). Wang pointed out that mapping a multi-dimensional abstract table into a two-dimensional database table requires determining which categories correspond to attribute names, primary keys, and non-primary keys.

Given data already stored in categories, a table designer can select any set of categories and combine them into a two-dimensional human-readable table. Conversely, given a set of two-dimensional tables, one may extract the categories and data. Our current work to facilitate queries on multiple tables builds on these observations.

2.3 Physical Structure Extraction (Low-level Table Processing).

In printed tables boxing, rules, or white space alignment are used for separating cell entries. Laurentini and Viada extracted cell corner coordinates from the ruling lines[11]. Image processing techniques for the extraction of physical structure from scanned tables include Hough Transforms[12], run-length encoding[13], word bounding boxes[14], and conditional random fields (CRFs)[15]. Hirayama segmented partially-ruled tables into a rectangular lattice[16]. Handley iteratively identified cell separators and successfully processed large, complex, fully-lined, semi-lined, and unruled tables with multiple lines of text per cell[17]. Zuyev identified cell contents for an OCR system using connected components and projection profiles[18]. The notion of converting paper tables into Excel spreadsheets dates back at least to 1998[19]. Early research in table processing suffered from the isolation of the graphics research community from the OCR community. Current OCR products can convert printed tables into a designated table format. Most desktop publishing software has provisions for the inter-

conversion of tables and spreadsheets. None of this software does explicitly retain header-value cell connections, but we believe that our methods can extend existing methods for physical segmentation of printed tables to logical analysis.

Less attention has been focused on ASCII[†] table analysis, where the structure must often be discovered from spacing, special symbols like "|" and "_", or the correlation of text blocks on successive lines. Pyreddy and Croft demonstrated results on over 6000 tables from the Wall Street Journal[20]. T-Recs clustered words for bottom-up structural analysis of ASCII tables[21]. Row and column alignment via directed acyclic attribute graphs was also explored[22]. Most of these methods could also be applied as a preprocessor for our algorithms. Research on ASCII tables has diminished since the development of XML for communicating structured data without sacrificing ASCII encoding.

The extraction of a web table's underlying grid structure, from its customary HTML representation, as opposed to extracting cell formatting, is relatively simple because the cells are already in row order. It is part of the *import* feature of most spreadsheet software. We use MS-Excel's.

2.4 Logical Structure Extraction (High-level Table Processing)

Gattebauer et al. presented a geometric approach to table extraction from arbitrary web pages based on the spatial location of table elements prescribed by the DOM tree[23]. They formulated a "visual table model" of nested rectangular boxes derived from *Cascading Style Sheets*. They applied spatial reasoning—primarily based on adjacency topology and Allen interval relations—to their visualization model in order to determine the final box structure. They also exploited semantic analysis with a known or assumed list of keywords. Their interpretation consists of XML-tagged generalized n-tuples. They evaluated several steps of their process on a set of 269 web pages with 493 tables and reported 48% precision with 57% recall. Allen's classification of the possible spatial relations of collinear line segments[24] has been generalized to an arbitrary number of dimensions[25]. We also use block relations to describe tables that fit our model.

Shamalian et al. demonstrated a model-based table reader for reading batches of similar tables[26]. Similarities between tables and forms were noted Bing et al. [27] and by Kieninger and Dengel[28] in the course of developing relevant image processing methods. Amano and Asada have published a series of papers on graph grammars based on box adjacency for "table-form" documents[29]. Their grammars encode the relationship between "indicator," "example," and "data boxes." Another group, headed by T. Watanabe, aimed at learning the various types of information necessary to interpret a ruled scanned table. They populated a "Classification Tree" from a training set of diverse tables. The nodes of the tree are "Structure Description Trees" that can interpret a specific family of tables. In the operational phrase, new classification nodes and tree structure descriptions are added for unrecognized tables[30]. Their model specifies the location of the data cells, thus obviating the need to interpret headers either syntactically or semantically. General grammar-based approaches that can be specialized to forms and tables have been demonstrated on large data sets by Coüasnon and his colleagues[31].

A series of papers culminating in V. Long's doctoral thesis[32] analyzes a large sample of tables from Australian Stock Exchange financial reports. An interesting aspect of this work is the detection and verification of the scope and value of *aggregates* like totals, subtotals, and averages. The analysis is based on a blackboard framework with a set of cooperating agents. This dissertation has a good bibliography of table papers up to 2009. More recently, Astrakhantsev also explored aggregates in tables[33].

Already in 1997, Hurst and Douglas advocated converting tables into relational form: "Once the relational structure of the table is known it can be manipulated for many purposes."[34]. Hurst provided a useful taxonomy of category attributes in terms of *is-a, part-of, unit-is,* and *quantity-is*. He pointed out that the physical structure of a table is somewhat analogous to syntax in linguistic objects. He also emphasized the necessity and role of natural language analysis for table understanding, including the syntax of within-cell strings[35]. Hurst's dissertation contains a wealth of interesting examples of tables[36]. Hurst's work was reviewed and extended by Costa e Silva et al., who analyzed prior work in terms of contributions to the tasks of *table location*, *segmentation*, *functional analysis* (tagging cells as data or attribute), *structural analysis* (header index identification), and *interpretation* (semantics)[37],. Costa e Silva also provides a clear distinction between tables, forms, and lists. The ultimate objective of this group is the operational analysis of financial tables with feedback between the five tasks based on confidence levels.

Kim and Lee reviewed web table analysis from 2000 to 2006 and found logical hierarchies in HTML tables using cell formats and syntactic coherency[38]. They extracted the table caption and divide spanning cells correctly. In contrast to many researchers, they handled vertical and horizontal column headers symmetrically.

[†] We use "ASCII" loosely as a coding convention for strings of symbols. It could be unicode or Guo Biao (GB) code for Chinese.

The TARTAR (*Transforming ARbitrary TAbles into fRames*) system developed by Pivk et al. has objectives similar to ours.[39]. In the cited paper, the authors demonstrated their work on HTML tables. Their analysis and region recognition was based on cell formats (letters, numerals, capitalization, and punctuation) rather than indexing properties. The cells were functionally labeled in a manner similar to Hurst as *access* or *data* cells and assembled into a *Functional Table Model*. An attempt was made to interpret strings semantically using WordNet. The final output was a semantic (F-logic) frame. The complex evaluation scheme that was presented and applied to 158 HTML tables was hampered by human disagreement over the description of the frames.

A team from Yahoo has formulated the ambitious notion of a *web of concepts* that goes beyond our current horizon[40]. We have not, however, found prior work that attempts to convert source tables to relational tables using row and column header paths and their intrinsic 2-D indexing properties.

In the last several years, an active and inventive group including Google researchers, possibly inspired by Halevy et al.[41], has harvested and analyzed millions of web pages containing <table> tags. Their general approach has been to treat table rows as tuples with attributes specified by the top row, called *schema*. Visual verification of their results has necessarily been restricted to much smaller samples[42,43]. Working with the corpus 154M schemas, Cafearella et al derived the attribute correlation statistics database (AcsDB) that records corpus-wide co-occurrence of schema elements[44].

Extending this work to tables more complex than simple relational tables, Adelfio and Samet leveraged the principles of table construction to generate interpretations for spreadsheet and HTML tables[45]. Using CRFs like Pinto et al.[15], they classified each row of a table as a header row, data row, title, blank row, etc. With their test set of 1048 spreadsheet tables and 928 HTML tables, they achieved an accuracy of 76.0% for classifying header and data rows for spreadsheet tables and 85.3% for HTML tables, and for classifying all rows, 56.3% and 84.6% respectively.

A recent paper by Chen and Cafarella[46] presented a table-processing system that transforms spreadsheet tables into relational database tables. Whereas our approach is algorithmic, they, (like Adelfio and Samet), adapt the CRF technique to label each row with one of four labels: title, header, data, and footnote, using similar row features. Their rows labeled as "data" also include the cells in the row header, hence to distinguish between the two, they must assume that the data region is purely numeric. Their hierarchy extractor builds ParentChild candidates of cells in the header region using formatting, syntactic, and layout features. The candidate list is pruned by an SVM classifier that forces the resulting set of candidate pairs to be cycle-free. In our algorithmic approach, the resulting structure is guaranteed to be cycle-free by construction. We note also that all of the above methods are all oblivious to category structure.

2.5 Our earlier work

Our collective work on tables includes surveys of table processing; collections of tables that stretch the very definition of *table*; disquisitions on the differences between tables, lists and forms; examples of human ambiguity in table interpretation; an exploration of the extent of semantic information revealed by table structure; (the notion of a Web-of-Knowledge (WoK) that is similar to the Yahoo researchers' *web-of-concepts*; matching input tables with known conceptualizations in an attempt to interpret them); s*ibling tables* for information extraction from batches of tables with similar headers; a taxonomy based on the geometric relationship of tabular structures to isothetic tessellations and to X-Y trees; and machine learning for table segmentation based on appearance features. However, our only papers directly related to the next two sections are those on factoring header paths[47], on algorithmic table segmentation based on the fundamental indexing property[48]; on transformation of human-readable tables into canonical tables for SQL queries and RDFs for SPARQL queries[49]; and on VeriClick, an interactive tool for table segmentation using critical cells[50].

3. METHOD

We first describe the types of tables that are accepted and processed by our programs. Then we describe each of the processing steps. Give the available space, we opted for more illustrations at the expense of formal proofs.

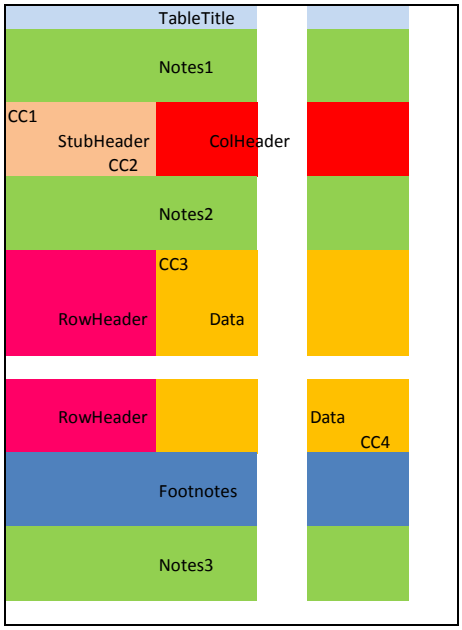
3.1 Well Formed Tables

Our programs process Well Formed Tables (WFTs), with a structure illustrated by the schematic diagram of Fig. 1a. The principal constraints are that the RowHeader must be to the left and aligned with the Data region, and that the ColumnHeader must be above and aligned with the Data region. The TableTitle is in the top row (usually a merged cell in HTML). FootnotePrefixes (like "*"), followed by the Footnote itself, must be below the RowHeader and the Data and cannot share their row with anything else. The corresponding FootnoteMarker may occur in any cell above the Footnote. Notes, which often provide information about the source or dissemination of the data, may occur above or below the

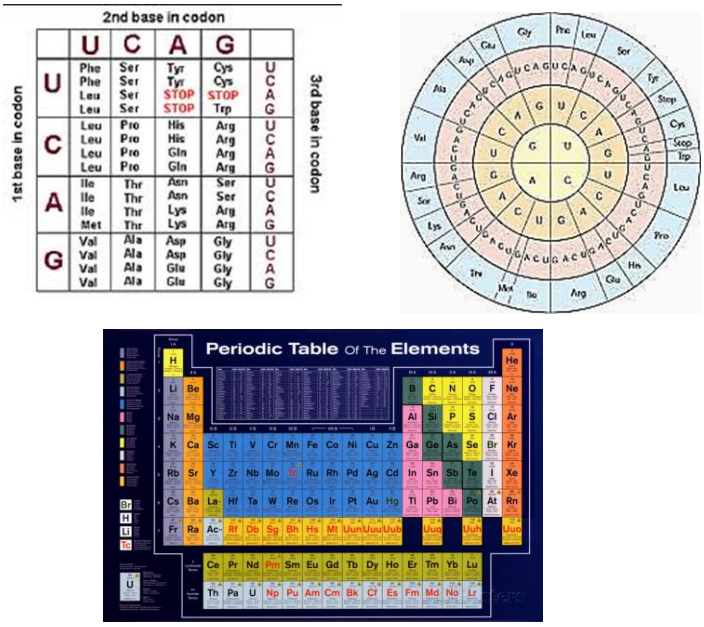
ColumnHeader or below the Footnotes. Empty rows or columns may occur anywhere except at the top or left (a CSV requirement), but are most common on the far right or below the table (they can be deleted without loss of information).

Fig. 1b displays three esoteric tables that do not follow the requirements of a Well Formed Table and that we cannot segment without first transforming them (so far, only manually) into WFTs. We also cannot yet process nested tables, concatenated tables, and tables with graphical cell contents.

The structure of WFTs can be formalized with block interval algebra, but we believe that Fig. 1 is an unambiguous and easy-to-grasp representation of our requirements for acceptable table layouts. The segmentation depends on the determination of four critical cells. CC1 and CC2 delimit the row and column headers, while CC3 and CC4 reveal the extent and alignment of the data region. The critical cells address only table geometry and topology. The logical aspects of indexing and category structure are discussed in Section 3.2 and 3.3.



(a)



(b)

Fig. 1. (a) Schematic diagram of a Well Formed Table; (b) Examples of esoteric tables that we cannot yet process.

3.2 Segmentation and Classification

Consider the tables of Fig. 2, as they might appear on the web or in Excel in XLSX format. Fig. 3 shows the rendering of the same tables after conversion to CSV format. Most of the formatting is lost, including cell size, colors, fonts, and text positioning within the cells. The conversion reveals that the table title is actually part of the table, as is the case with most web tables that we have seen. Although Excel displays the CSV file as a table, with left-justified text and right-justified numerals, the Notepad display of Fig. 4 shows that the CSV file contains only commas and EOLs as cell content separators. Fig. 5 shows how we fill the elementary cells generated from merged cells by conversion to CSV.

Table *12a*. Agricultural Production

	2010		2011	
	Egypt	Libya	Tunisia	Algeria
Wheat	11,000	3,000	7,400	3,800
Corn	8,000	5,500	6,950	4,340

Table *12b*. Agricultural Production

		Wheat	Corn
Egypt	2010	11,000	8,000
	2011	7,400	6,950
Libya	2010	3,000	5,500
	2011	3,800	4,340

Table *12c*. Agricultural Production

	2010		2011	
	Egypt	Libya	Egypt	Libya
Wheat	11,000	3,000	7,400	3,800
Corn	8,000	5,500	6,950	4,340

Table *12d*. Agricultural Production

		Egypt	Libya
Wheat	2010	11,000	3,000
	2011	7,400	3,800
Corn	2010	8,000	5,500
	2011	6,950	4,340

Fig. 2. Four tables rendered before conversion to CSV format.

Table 12a. Agricultural Production				
	2010		2011	
	Egypt	Libya	Tunisia	Algeria
Wheat	11000	3000	7400	3800
Corn	8000000	5500	6950	4340

Table 12b. Agricultural Production			
		Wheat	Corn
Egypt	2010	11000	8000
	2011	7400	6950
Libya	2010	3000	5500
	2011	3800	4340

Table 12c. Agricultural Production				
	2010		2011	
	Egypt	Libya	Egypt	Libya
Wheat	11000	3000	7400	3800
Corn	8000	5500	6950	4340

Table 12d. Agricultural Production			
		Egypt	Libya
Wheat	2010	11000	3000
	2011	7400	3080
Corn	2010	8000	5500
	2011	6950	4340

Fig. 3. The same tables after conversion to CSV format

```
Table 12d.  Agricultural Production,,,
,,Egypt,Libya
Wheat,2010,11000,3000 , n = 1
,2011,7400,3080
Corn,2010,8000,5500
,2011,6950,4340
```

Fig. 4. Notepad display of Table 12d in Fig; 4.

Table 12c. Ag	Table 12c. Ag	Table 12c. Ag	Table 12c. Ag	Table 12c. Ag
BLANC	2010	2010	2011	2011
BLANC	Egypt	Libya	Egypt	Libya
Wheat	11000	3000	7400	3800
Corn	8000	5500	6950	4340

Fig. 5. Table 12c after refilling cell contents.

The extraction of row and column headers is based on the recursive MIPS algorithm presented at ICDAR 2013[47] which finds the critical cell CC3. For Table 12a in Fig. 3, the minimal column header that we extract consists of only the third row of the table, because the labels Egypt, Libya, Tunisia and Algeria form unique column header paths. For Table 12c, the second row is also required because of the repetitive labels in the third row. Table 12d has a single-row column header, but a two-column row-header. (Multi-row row/column headers are more common than multi-column row headers.) The program also finds empty rows or rows containing repetitive units above the data cells, and rows containing footnotes or other notes below the data cells. The segmentation results are saved in a CSV *Classification Table*, such as that shown in Fig. 6 for Table 12c.

3.3 Category Extraction

Table 12a in Figs. 2 and 3 is a two-category table. The others are three-category tables because either the column header or the row header can be factored into a cross product:

Table 12b, row header (transposed): (Egypt*2010)+(Egypt*2011)+(Libya*2010)+(Libya*2011)
= (Egypt+Libya)*(2010+2011)

Table 12c, column header: (2010*Egypt)+(2010*Libya)+(2011*Egypt)+(2011*Libya)
= (2010+2011)*(Egypt +Libya)

Table 12d row header (transposed): (Wheat*2010)+(Wheat*2011)+(Corn*2010)+(Corn *2011)
= (Wheat + Corn)*(2010+2011)

The asterisk in the column header or transposed row header stands for vertical concatenation, and the plus sign for horizontal adjacency. The factorization uses only the associative and distributive laws of algebra. The results of the factorization are saved in a CSV *Category Table* with one row for each data cell, as seen in Fig. 7 for Table 12c. For real web tables, both of these CSV tables typically contain several hundred rows. The number of columns in the Classification Table is always five, while the Category Table typically has only four columns (only 10 % of our tables have more than two categories), of which all but the last serve as a composite *key* to some data.

3.4 Distance Computation and Clustering

The features used for determining the similarity or distance of two categories (from different tables) are just the unique words in each category header (or in the table title). The words in the category header are extracted by sweeping the Category Table and retaining only unique words, while the words in the table title are extracted from the Classification Table. The original CSV table is never invoked. The net result of this step is a WordSet where each sublist of unique words is indexed by a tableID and the specific category label (RowCat_1, RowCat_2, ColCat_1, …) or TableTitle. Because each

table has at least a row category, a column category, and a table title, the Word List contains at least three times as many sublists as there are input tables. The WordSet for tables 12a and 12c is displayed in Fig. 8.

Classification Table

Cell_ID	Row	Col	Content	Class
12c_R1_C1	1	1	Table 12c. Agricultural Pr	tabletitle
12c_R1_C2	1	2		tabletitle
12c_R1_C3	1	3		tabletitle
12c_R1_C4	1	4		tabletitle
12c_R1_C5	1	5		tabletitle
12c_R2_C1	2	1		stubheader
12c_R2_C2	2	2	2010	colheader
12c_R2_C3	2	3	2010	colheader
12c_R2_C4	2	4	2011	colheader
12c_R2_C5	2	5	2911	colheader
12c_R3_C1	3	1		stubheader
12c_R3_C2	3	2	Egypt	colheader
12c_R3_C3	3	3	Libya	colheader
12c_R3_C4	3	4	Egypt	colheader
12c_R3_C5	3	5	Libya	colheader
12c_R4_C1	4	1	Wheat	rowheader
12c_R4_C2	4	2	11000	data
12c_R4_C3	4	3	3000	data
12c_R4_C4	4	4	7400	data
12c_R4_C5	4	5	3800	data
12c_R5_C1	5	1	Corn	rowheader
12c_R5_C2	5	2	8000	data
12c_R5_C3	5	3	5500	data
12c_R5_C4	5	4	6950	data
12c_R5_C5	5	5	4340	data

Fig. 6 (left) Classification Table for Table 12c. Whereas the Category Table covers only Data cells, this Classification Table retains all the content information about the input table. The Classification Table can be inverted to reproduce the input table, or read directly into a relational database or OWL collection. In addition to the five classes shown here, it can designate Footnote Prefix, Footnote, Footnote Marker, Notes, and Empty rows or columns.

Category Table

Cell_ID	RowCat_1	ColCat_1	ColCat_2	Data
12c_R4_C2	Wheat	2010	Egypt	11000
12c_R4_C3	Wheat	2010	Libya	3000
12c_R4_C4	Wheat	2011	Egypt	7400
12c_R4_C5	Wheat	2011	Libya	3800
12c_R5_C2	Corn	2010	Egypt	8000
12c_R5_C3	Corn	2010	Libya	5500
12c_R5_C4	Corn	2011	Egypt	6950
12c_R5_C5	Corn	2011	Libya	4340

Fig. 7 (above) Category Table for Table 12c. This is a relational table that can be read directly into Access or into an a collection of RDF triples for query formulation..

T12a	tabletitle	'Table', '12c.', 'Agricultural' 'Production'
T12a	RowCat_1	'Wheat', 'Corn'
T12a	ColCat_1	'Egypt', 'Libya', 'Tunisia', Algeria'
T12c	tabletitle	'Table', '12c.', 'Agricultural' 'Production'
T12c	RowCat_1	'Wheat', 'Corn'
T12c	ColCat_1	'2010', '2011'
T12c	ColCat_2	'Egypt', 'Libya'

Fig. 8. WordSet (of unique words) of the table titles and category headers of Table 12a and 12c.
Note that 20010, 2011 are not part of the minimal column header of T12a because all the labels in the next row are unique.

Our measure of dissimilarity is the Jaccard distance DJ(p,q) between two sets of unique words p and q:

$$D_J(p,q) = 1 - |p \cap q| / |p \cup q|.$$

D_J is a proper metric:

$$D_J(p,p) = 0; \quad D_J(p,q) = D_J(q,p); \quad 0 \leq D_J(p,q); \quad D_J(p,q) + D_J(q,r) \leq D_J(p,r).$$

For our example, $D_J(\{T12a; ColCat\_1\}, \{T12c; ColCat\_2\}) = 2/4 = 0.5$. The Jaccard distance is computed for every pair of rows in WordSet and saved as a (large) CSV file. Fig. 11 in Section 4.2 is an example of a partial Distance Table. We proceed at follows:

1. Convert cell strings corresponding to the table title and the header category labels to sets of unique words appearing in each string. Let $W_1,\ldots, W_m$ represent these sets of words.

2. Compute the $D_J(W_i, W_j)$ for i, = 1, …, m, j, …, i-1, between W_i and Wj.

Note: A 1 value of the distance means that the two sets are disjoint and a 0 value means that the two sets are *identical*.

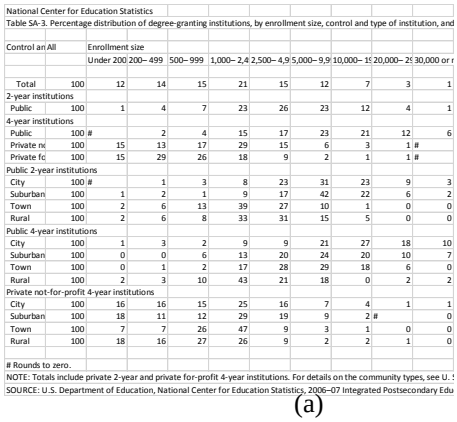
Our sequential similarity clustering algorithm is barely different from the one proposed by Hall in 1967[9]. Each sample is compared, in a preset order, to all samples in every already existing cluster. If the minimum distance it less than θ_{LOW}, than the new sample is assigned to the cluster of the nearest previously assigned sample. If the distance to every previous sample is greater than θ_{HIGH}, then it becomes the seed of a new cluster. Otherwise the sample is discarded and will never be assigned to any cluster.. Slightly more formally:

Let the samples be S_j, j = 1 to m, and the clusters be C_k, k = 1 to n. Initialization: $S_1 \rightarrow C_1$, n = 1, Dmin = 1
For k = 2 to m # for every sample, in some preset order
    For c = 1 to n # for every cluster
        For j = 1 to $|C_c|$ # $|C_c|$ is the number of samples in Cluster C_c
            If $D_J(S_k, S_{i(j)}) < D_{min}$,
                $D_{min} = D_J(S_k, S_{i(j)})$ and C =c # keep track of cluster with nearest sample
    If Dmin, < θ_{LOW}, $S_k \rightarrow C_C$; # assign sample to cluster with nearest sample
    If D_{min}, > θ_{HIGH}, $S_k \rightarrow C_{n+1}$ and n = n+1 # create a new cluster with only this sample
End

Clearly the method is order dependent. There are, however, few clustering algorithms that do not depend on order or initial seeds. We built in a random permutation to vary the order, and do plan to experiment with it as well as with the usual iterative enhancements to assign the discarded samples to clusters.

4. EXPERIMENTAL RESULTS

Our (small) sample consists of 200-tables drawn from ten institutional statistical sites like Statistics Canada and Norway Statistics. The only ground truth we have is the location of the four critical cells of the CSV version of each table. Figure 9 shows some examples, necessarily very small, of the original web tables.



National Center for Education Statistics

Table SA-3. Percentage distribution of degree-granting institutions, by enrollment size, control and type of institution, and

Control an	All	Enrollment size								
		Under 200	200– 499	500– 999	1,000– 2,4	2,500– 4,9	5,000– 9,9	10,000– 1	20,000– 2	30,000 or
Total	100	12	14	15	21	15	12	7	3	1
2-year institutions										
Public	100	1	4	7	23	26	23	12	4	1
4-year institutions										
Public	100	#	2	4	15	17	23	21	12	6
Private n	100	15	13	17	29	15	6	3	1	#
Private fo	100	15	29	26	18	9	2	1	1	#
Public 2-year institutions										
City	100	#	1	3	8	23	31	23	9	3
Suburban	100	1	2	1	9	17	42	22	6	2
Town	100	2	6	13	39	27	10	1	0	0
Rural	100	2	6	8	33	31	15	5	0	0
Public 4-year institutions										
City	100	1	3	2	9	9	21	27	18	10
Suburban	100	0	0	6	13	20	24	20	10	7
Town	100	0	1	2	17	28	29	18	6	0
Rural	100	2	3	10	43	21	18	0	2	2
Private not-for-profit 4-year institutions										
City	100	16	16	15	25	16	7	4	1	1
Suburban	100	18	11	12	29	19	9	2	#	0
Town	100	7	7	26	47	9	3	1	0	0
Rural	100	18	16	27	26	9	2	2	1	0

Rounds to zero.

NOTE: Totals include private 2-year and private for-profit 4-year institutions. For details on the community types, see U.

SOURCE: U.S. Department of Education, National Center for Education Statistics, 2006–07 Integrated Postsecondary Edu

(a)

People employed, by educational attainment

	2013		
	Both sexes	Men	Women
	%		
Total	**61.8**	**65.8**	**58.0**
15 to 24 years	55.1	54.2	56.0
25 to 44 years	81.9	85.8	77.9
45 and over	51.2	56.3	46.5
Less than Grade 9	19.8	27.8	12.7
15 to 24 years	23.8	27.6	19.2
25 to 44 years	50.5	65.3	32.0
45 and over	16.0	22.8	10.4
Some secondary school	39.5	46.0	32.3
15 to 24 years	35.6	36.0	35.1
25 to 44 years	63.5	71.6	50.8
45 and over	34.6	44.1	25.9

1. Includes trades certificate.
Source: Statistics Canada, CANSIM, table 282-0004 and Catalogue no. 89F0133XIE.
Last modified: 2014-01-10.

(b)

Fig. 9. Samples of web tables used in our experiments. Table (b) has two-category row headers.

4.1 Segmentation and Category Extraction

The segmentation program accepted only 198 of the 200 tables because two tables have duplicate rows or columns. Of the remaining 198 tables, one showed a discrepancy with the ground truth, as seen in the output of the segmentation program in Fig. 10. In the questionable Table 24 (cf. Fig. 9a) the program accepted the unique blank cell as a column header path, but the ground truth insisted on appending the row above it, resulting in the path ['All', ' '].

TableID	**CC1**	**CC2**	**CC3**	**CC4**	**TableID**	**CC1**	**CC2**	**CC3**	**CC4**	
…										
C10021.csv	A2	A2	B3	J18	C10021.csv	A2	A2	B3	J18	--
C10022.csv	A2	A3	B4	K26	C10022.csv	A2	A3	B4	K26	--
C10023.csv	A2	A2	B3	F16	C10023.csv	A2	A2	B3	F16	--
C10024.csv	A5	A5	B7	K28	C10024.csv	A4	A5	B7	K28	ERROR
C10025.csv	A4	A4	B5	F26	C10025.csv	A4	A4	B5	F26	--
C10026.csv	A4	A5	B6	E24	C10026.csv	A4	A5	B6	E24	--
C10027.csv	A4	A4	B5	F22	C10027.csv	A4	A4	B5	F22	--
…										

Fig. 10 CC_OUT. Partial output of the segmentation program and of the corresponding ground truth on the right.

Altogether 30,490 table cells were classified. The factorization revealed that 6 row headers and 15 column headers have more than one category. The segmentation and classification of 200 tables takes our unoptimized Python 2.7 program 12 seconds on a 2.4 GHz desktop. The distance computation and clustering add 3 seconds.

4.2 Distance Computation and Clustering

The WordSet for the 198 tables has 615 rows (3 × 198 for all tables + 6 for the two-row-category- tables + 15 for the two-column-category tables), with a total of 7475 words unique to each row. Therefore 615 × 615 = 378,225 distances were computed. Fig. 11 shows the partial Jaccard Distance table. It is, of course, symmetric, with zeroes on the diagonal. 217 distinct pairs of entries also have 0 distance because they have the same words. In contrast, there are 138,393distinct pairs that do not share any word ($D_J = 1$). There are 19,367 distinct pairs that are "similar," ($D_J \leq 0.5$).

Table I shows some results of the clustering program. The average membership of the multi-member clusters barely changes with an eight-fold increase in the number of such clusters, and with given thresholds the memberships were stable under permuted orders of presentation. When the two thresholds are equal, all of the word strings are assigned to some category. Tight clusters of category headers generally mean that the source tables can be combined along some dimension.

		C10001	C10001	C10001	C10001	C10002	C10002	C10002	C10003	C10003	C10003
		tabletitle	RowCat_1	ColCat_1	ColCat_2	tabletitle	RowCat_1	ColCat_1	tabletitle	RowCat_1	ColCat_1
C10001	tabletitle	0	1	0.9	0.909091	0.96	0.967742	1	0.875	0.968254	1
C10001	RowCat_1	1	0	1	1	1	1	1	1	1	1
C10001	ColCat_1	0.9	1	0	1	0.965517	1	1	0.952381	0.938462	1
C10001	ColCat_2	0.909091	1	1	0	1	1	1	0.909091	0.982759	1
C10002	tabletitle	0.96	1	0.965517	1	0	1	1	0.96	0.986111	1
C10002	RowCat_1	0.967742	1	1	1	1	0	1	0.967742	1	1
C10002	ColCat_1	1	1	1	1	1	1	0	1	1	1
C10003	tabletitle	0.875	1	0.952381	0.909091	0.96	0.967742	1	0	0.968254	1
C10003	RowCat_1	0.968254	1	0.938462	0.982759	0.986111	1	1	0.968254	0	1
C10003	ColCat_1	1	1	1	1	1	1	1	1	1	0
C10004	tabletitle	0.954545	1	1	1	0.52381	0.972222	1	0.904762	1	1
C10004	RowCat_1	0.967742	1	1	1	1	0	1	0.967742	1	1
C10004	ColCat_1	1	1	1	1	1	0.965517	1	1	1	1
C10005	tabletitle	0.95	0.944444	1	1	1	0.970588	1	0.95	1	1
C10005	RowCat_1	1	1	1	1	1	1	1	1	1	1
C10005	ColCat_1	1	1	0.947368	1	1	0.965517	1	1	0.983871	1
C10006	tabletitle	1	1	1	1	0.96	1	1	0.941176	1	1
C10006	RowCat_1	1	1	1	1	1	1	1	1	1	1
C10006	ColCat_1	1	1	1	1	1	1	1	1	1	1

Fig. 11. Distance Table (partial output).

Table I. Cluster membership vs. thresholds.

θ_{LOW}	0.00	0.05	0.05	0.50
θ_{HIGH}	1.00	0.95	0.05	0.50
Number of multi-member clusters	9	11	52	72
Samples in multi-member clusters	33	49	155	290
Number of single-member clusters	50	86	460	325

At small values of θLOW, most of the clusters consisted of identical headers or titles. For example, the cluster

C10001_RowCat_1 C10008_ColCat_1 C10073_RowCat_1 C10080_ColCat_1

had identical row or column headers of 2002 2003 2004 2005 2006 2007 2008. With a higher value θ_{LOW}, this cluster grew to 28 headers, all showing years, such as 2000 2001 2002 2003 2004 2005 2006 2007 2008 and 2006 2007 2008 2009.

An example of a cluster of non-identical table titles is:

TABLE 4. Port Calls By Vessel Type, Port of New York/New Jersey: 2007
TABLE 4. Port Calls by Vessel Type, Port of Charleston, SC: 2007
TABLE 4. Port Calls By Vessel Type, Port of New York, 2003
TABLE 4. Port Calls By Vessel Type, Port of Philadelphia, PA: 2007
TABLE 4. Port Calls By Vessel Type, Port of Los Angeles, CA: 2008
TABLE 4. Port Calls by Vessel Type, Port of Houston, TX: 2007
TABLE 4. Port Calls By Vessel Type, Port of Los Angeles, 2003
TABLE 4. Port Calls By Vessel Type, Port of Norfolk, VA: 2007
TABLE 4. Port Calls By Vessel Type, Port of Long Beach, CA: 2008

Identical titles don't necessarily mean the same content. For example, in a cluster of three identical top rows,

Renewable Energy Trends in Consumption and Electricity, 2007

We find the following column headers:

Company Name	Plant I.D.	Plant Name	County	Biomass/ Coal Cofiring Capacity	Total Plant Capacity

2003	2004	2005	2006	2007

Biomass			Geothermal	Hydroelectric Conventional	Solar/PV	Wind	Total
Waste		Wood and Derived Fuels3					
Landfill Gas	MSW Biogenic1						

This is actually an error in locating the table title: the actual titles are in the fifth row, with the first row containing only the title of the shared source report. The titles do not, however, cluster, and therefore would not show, without deep semantic analysis, that these tables are closely related. A more embarrassing error found by clustering was the occurrence of two identical tables in our dataset. From our ever optimistic view, even these errors show the value of clustering headers for improving other table analysis tasks. We have not yet attempted any quantitative means of evaluating these results using queries.

5. DISCUSSION

We presented a straightforward methodology for (1) extracting unique words from header categories and table titles, (2) computing a distance function between header categories, and (3) clustering the header categories and table title. These procedures build directly on the relational tables produced by our previous work on table segmentation, header category factoring, and cell classification. In fact, the python code that accomplishes the new table analysis functions requires only the Classification and Categorization tables and never reads the original tables that were imported from the web.

In text categorization it is common to augment the words of a document text by synonyms and to eliminate stop words. Tables are, however, far terser than narrative text: the number of unique words in a table title and headers is over two orders of magnitude smaller than in a web page of text. As Adelfio and Samet put it: *…due to the semantic meaning communicated by their layout and structure, the need for descriptive words is minimized, allowing tables to communicate more information than prose in the same amount of space*[45]. The average fraction of words in our table titles for which we found synonyms using by the Python natural language tool box (http://www.nltk.org/book/) is 55.2%. It was. 46.6% for row and column headers. Many of these, however, are not likely to be useful for finding related tables: “two” or “deuce” for “2”, or “”hi”, “me”, ‘in”, “or”, “ok” for US state names. As regards stop words, they constitute only 20.1% of our table titles and 4.1% of category headers. Discarding acronyms like “OPEC” is also questionable because due to the limited space in the table they are often expanded only in the text surrounding the table.

The Jaccard distance is the accepted measure for comparing unordered sets of varying size. Perhaps a more interesting question is the choice of clustering algorithm. We make no claim of any contribution for our choice or implantation. Sound selection depends on the nature of the data, the number of samples, the features available to characterize the samples and, significantly, on the purpose of the grouping procedure. The final choice normally calls for a great deal of experimentation: it is, in fact, often called *exploratory data analysis*. To demonstrate an application of a similarity measure over category headers, we simply chose the simplest and fastest algorithm that we knew. As one of a small family that requires only a single pass over the data, its appeal is that it can be easily scaled to tens, if not hundreds, of thousands of tables (on a desktop rather than a computer farm). We note that the developers of NewsStand[51] made the same choice for news items.

We look forward to reporting at a future DR&R queries based on the results of our just-completed clustering procedure. For now the most we can say is that extracting and clustering category headers offers new research opportunities in table recognition and retrieval.

ACKNOWLEDGMENT

Mukkai acknowledges the help of Dr. Ravi Palla with Protegé.

REFERENCES

[1] J. McQueen "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability,* 281-297, 1967.

[2] G. H. Ball, D.J. Hall, . "ISODATA, a novel method of data analysis and classification," Tech. Rep. Stanford University, Stanford, CA, 1965.

[3] J.A. Hartigan, *Clustering Algorithms*, John Wiley & Sons, New York, 1975.

[4] R. C. Dubes, A.K. Jain, "Clustering techniques: The user's dilemma," in Pattern Recognition 8, 247-260, 1976.

[5] A.K. Jain, M.N. Murty, P.J. Flyinn, "Data Clustering, a review," in ACM Comp. Surveys, 31, 3, 264-323, 1999.

[6] S. Theodoridis, K. Koutroumbas, *Pattern Recognition*, Elsevier, 2009.

[7] A. Kontostathis, W.P. Pottenger, "A framework for understanding Latent Semantic Indexing (LSI) performance," in *Information Processing & Management*, 42, 1, 56-73, 2006.

[8] S. Deerwester, et al. "Indexing by latent semantic analysis," in *J. Am. Soc. Information Science*, 41,6, 391-407, 1990.

[9] A.V. Hall,"Methods for demonstrating resemblance in taxonomy and ecology," in *Nature* 214, 830-831, 1967.

[10] X. Wang, *Tabular abstraction, editing, and formatting*, Ph.D. thesis, University of Waterloo, Canada, 1996.

[11] A. Laurentini, P. Viada, "Identifying and understanding tabular material in compound documents," in:*Procs. 11th ICPR (ICPR'92),* 405–409, The Hague, 1992.

[12] E. Turolla, Y. Belaid, A. Belaid, "Form item extraction based on line searching," in Kasturi, R., Tombre, K. (eds.) *Graphics Recognition—Methods and Applications*. LNCS 1072, 69–79. Springer-Verlag, Berlin, 1996.

[13] S. Chandran, R. Kasturi, "Structural recognition of tabulated data," in *Proceedings of the Second International Conference on Document Analysis and Recognition* (ICDAR'93), 516–519. Tsukuba Science City, Japan, 1993.

[14] K. Itonori, "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'93), 765–768. Tsukuba Science City, Japan,1993.

[15] D. Pinto, A. McCallum, X. Wei, W.B. Croft, "Table extraction using conditional random fields," in *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 235–242, 2003.

[16] Y. Hirayama, "A method for table structure analysis using DP matching," in *Proceedings of the Third International Conference on Document Analysis and Recognition* (ICDAR'95), 583– 586. Montréal, Canada, 1995.

[17] J.C. Handley, "Document recognition," in Dougherty, E.R. (ed.) *Electronic Imaging Technology*, Chapter 8. SPIE—The International Society for Optical Engineering, 1999.

[18] K. Zuyev, "Table image segmentation, " in *Proceedings of the International Conference on Document Analysis and Recognition* (ICDAR'97), 705–708, 1997.

[19] A. Abu-Tarif, *Table processing and table understanding*, Rensselaer Polytechnic Institute MS thesis, 1998.

[20] P. Pyreddy,W.B. Croft, *TINTIN: A system for retrieval in text tables*, Technical Report UM-CS-1997-002, University of Massachusetts, Amherst, 1997.

[21] T.G. Kieninger, "Table structure recognition based on robust block segmentation," in *Proceedings of Document Recognition V* (IS&T/SPIE Electronic Imaging'98), vol. 3305, 22–32. San Jose, CA, 1998.

[22] J. Hu, R. Kashi, R, D. Lopresti, G.Wilfong, "Table structure recognition and its evaluation," in: Kantor, P.B., Lopresti, D.P., Zhou, J.(eds.) *Proceedings of Document Recognition and Retrieval* VIII (IS&T/SPIE Electronic Imaging), vol. 4307, 44–55. San Jose, CA,2001.

[23] W. Gatterbauer, P. Bohunsky, M. Herzog, B. Krupl, B. Pollak, "Towards Domain-Independent Information Extraction from Web Tables," in *Proceedings of the 16th International Conference on World Wide Web*, Banff, Alberta, Canada, 71-80, 2007.

[24] J.F. Allen, "Maintaining knowledge about temporal intervals," in *Comm. ACM* 26, 11, Nov. 1983.

[25] P. Balbiani, J.F. Condotta, L. Farinas del Cerro, "Tractability results in the Block Algebra," in *J. Logic. Computat*. 12, 5, 885-909, 2001.

[26] J.H. Shamalian, H.A. Baird, R.L. Wood, "A retargetable table reader," in *Proceedings of the International Conference on Document nalysis and Recognition* (ICDAR'97), 158–163, 1997.

[27] L. Bing, J. Zao, X. Hong,"New method for logical structure extraction of form document image," in *Proceedings of Document Recognition and Retrieval VI* (IS&T/SPIE Electronic Imaging' 99), vol. 3651, 183–193. San Jose, CA, 1999.

[28] T. Kieninger, A. Dengel, "A paper-to-HTML table converting system," in *Proceedings of Document Analysis Systems* (DAS) 98. Nagano, Japan, 1998.
[29] A. Amano, N. Asada, "Graph grammar based analysis system of complex table form document," in *Proceedings of the Seventh International Conference on Document Analysis and Recognition*, Edinburgh 2003.
[30] T. Watanabe, Q.L. Quo, N. Sugie,"Layout recognition of multikinds of table-form documents," *IEEE Trans. Pattern Anal. Mach. Intell*, 17(4),432–445, 1995.
[31] A. Lemaitre, J. Camillerapp, B. Coüasnon. "Multiresolution Cooperation Improves Document Structure Recognition,". in *Int. J. Doc. Anal. Recognit*. (IJDAR), Springer, 11(2):97-109, 2008.
[32] V. Long, *An Agent-Based Approach to Table Recognition and Interpretation*, Macquarie University PhD dissertation, May 2010.
[33] N. Astrakhantsev, "Extracting Objects and Their Attributes from Tables in Text Documents," in Denis Turdakov, Andrey Simanovsky (Eds.*): Proceedings of the Seventh Spring Researchers Colloquium on Databases and Information Systems*, SYRCoDIS, Moscow, Russia, June 2-3, 2011 34-37, 2011.
[34] M. Hurst, S. Douglas, "Layout and language: Preliminary investigations in recognizing the structure of tables." in *Proceedings of the Int. Conference on Document Analysis and Recognition* (ICDAR'97), 1043–1047, 1997.
[35] M. Hurst, "Towards a theory of tables," in *Int. J. Doc. Anal. Recognit*. 8 (2-3), Springer, 66-86, 2006.
[36] M. Hurst, *The Interpretation of Tables in Texts*. Ph.D. thesis, University of Edinburgh, 2000.
[37] A. Costa e Silva, A. M. Jorge and L. Torgo, "Design of an end-to-end method to extract information from tables," in *Int. J. Doc. Anal. Recognit.* 8 (2-3), Springer, Heidelberg, 66-86, 2006.
[38] K. Yeon-Seok, L. Kyong-Ho "Extracting logical structures from HTML tables," in Computer *Standards & Interfaces*, 30 (5), 296-308 July 2008.
[39] A. Pivk, P. Cimiano, Y. Sure, M. Gams,V. Rajkovic, R. Studer, "Transforming arbitrary tables into logical form with TARTAR," in *Data & Knowledge Engineering* 60, 567–595, July 2008.
[40] N. Dalvi, R. Kumar, B.Pang, R. Ramakrishnan A. Tomkins, P. Bohannon, S. Keerthi, "A Web of Concepts," in *ACM SigMod PODS'09,* June 29–July 2, 2009, Providence, Rhode Island, USA, 2009.
[41] A. Halevy, P. Norvig, and F. Pereira, "The Unreasonable Effectiveness of Data," in *IEEE Intelligent Systems*, March/April 2009.
[42] P. Venetis, A. Halevy. J. Madhavan, M. Pasca, W. Shen, F. Wu, G. Miao, C. Wu, "Recovering Semantics of Tables on the Web," in *Proceedings of the LDB Endowment*, 4, (9), 2011.
[43] H. Gonzalez, A. Y. Halevy, C. S. Jensen, A. Langen, J. Madhavan, R. Shapley, W. Shen, J. Goldberg-Kidony, Google Fusion Tables: Web-Centered Data Management and Collaboration, *SIGMOD'10*, Indianapolis, Indiana,2010.
[44] W.J. Cafarella,A. Halevy, D.Z. Wang, E. Wu, Y. Zhang, "WebTables: Exploring the Power of Tables on the Web," in *Proceedings of the VLDB Endowment '08* Auckland, New Zealand, 2008.
[45] M.D. Adelfio, H. Samet, "Schema Extraction for Tabular Data on the Web", in *Proceedings of the VLDB Endowment*, 6, (6), Riva del Garda, Trento, Italy 26–30 August, 2013.
[46] Z. Chen and M. Cafarella, "Automatic Web Spreadsheet Data Extraction", in *Proceedings of the 3rd International Workshop on Semantic Search over the Web* (SSW 2013), Riva del Garda, Trento, Italy, 30 August 2013.
[47] D.W. Embley, M. Krishnamoorthy, G. Nagy, S. Seth, "Factoring Web tables," in *Procs. EIA/AIE Conf.* (F. Esposito, S. Ferilli, eds.), ACM, February 2011 and in LNAI 6703, p. 253-263, June 2011.
[48] S. Seth, and G. Nagy, "Segmenting Tables via Indexing of Value Cells by Table Headers," in *Proc. Int. Conference on Document Analysis and Recognition* (ICDAR'13), Washington, D.C., August 2013.
[49] D.W. Embley, S. Seth, G. Nagy, "Transforming Web tables to a relational database." in *Proceedings of the 22nd International Conference on Pattern Recognition* (ICPR'14), Stockholm, Sweden, August 2014.
[50] G. Nagy, M. Tamhankar, "VeriClick, an efficient tool for table format verification," in *Procs.Conf. on Document Recognition and Retrieval* (SPIE/EIT/DR&R), San Francisco, Jan. 2012.
[51] H.Samet et al., "Reading News with Maps by Exploiting Spatial Synonyms," *Comm. ACM* 57, 10, Nov. 2014.