Recognizing Records from the Extracted Cells of Microfilm Tables [*]

Kenneth M. Tubbs
David W. Embley
Department of Computer Science
Brigham Young University
Provo, UT 84602
Contact Author: David W. Embley, (801) 422-6470

{tubbs,embley}@cs.byu.edu

ABSTRACT

Microfilm documents contain a wealth of information, but extracting and organizing this information by hand is slow, error-prone, and tedious. As an initial step toward automating access to this information, we describe in this paper an algorithmic process to automatically identify record patterns found in microfilm tables. Our table-processing algorithm accepts an XML input file describing the individual cells of a table taken from a microfilm document, finds for each record in the document the cells that together comprise the record. Two key features drive the algorithm: (1) geometric layout and (2) label matching with respect to a given application ontology. The algorithm achieved an accuracy of 92% on our test corpus of genealogical microfilm tables.

Keywords: microfilm tables, automated recognition of record patterns, geometric layout, ontology matching.

1. INTRODUCTION

Several million rolls of microfilm exist, and each roll contains about a thousand images of data. Although of great interest to many people, this wealth of information is nearly inaccessible because extracting, checking, and indexing it by hand is extremely time consuming.[1] In this paper we make an initial step toward automating access to this information. Although only the beginning, the step we take is significant because it lays the ground work (1) for organizing and indexing the information and (2) for automatic (or semiautomatic) extraction of the data, which is mostly handwritten.

[*]This material is based upon work supported by the National Science foundation under Grant No. IIS-0083127.

[1][14] estimates that it would take 104 years for 20,000 extractors working 100 hours per year to extract the information from 2.5 million rolls of microfilm, the number of rolls held by a large genealogical library.

We are able to organize and index the information because we can identify records, determine which fields constitute a record, what each field is, and where each field is located in the image within the collection of images for a roll of microfilm. We are able to lay the groundwork for intelligent recognition of handwriting because we can determine the type of information in each cell.[2]

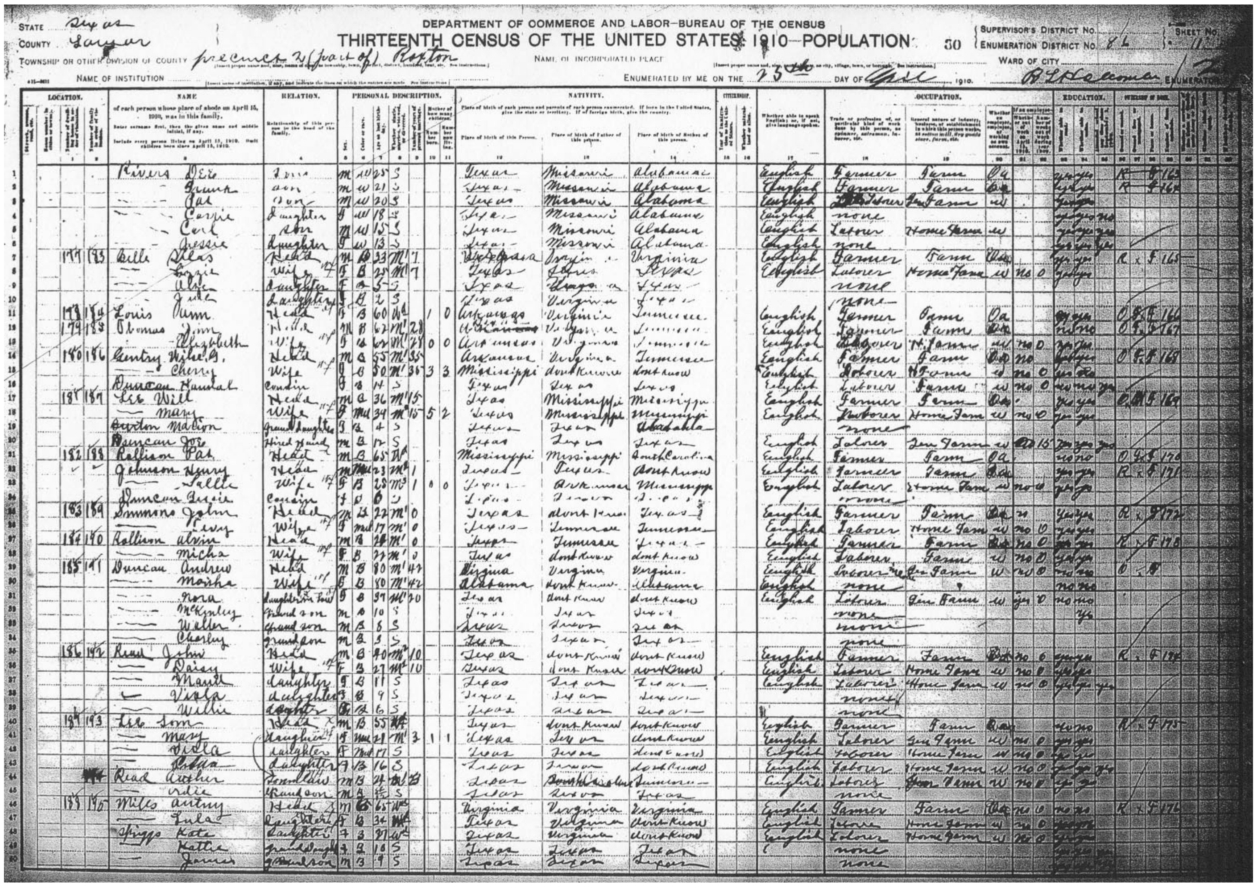
One type of microfilm document is a table with machine-printed labels and handwritten values. As an example, Figure 1 shows a page from the 1910 U.S. Census. This type of table appears as a rectangular grid of cells. A *cell* is a rectangle within the grid. A series of horizontally connected cells is a *row*, and a series of vertically connected cells is a *column*. Tables contain three different types of cells: (1) *label cells*, which contain machine-printed text, (2) *value cells*, which contain handwritten data described by a label cell, and (3) *empty cells*, which are empty, but may contain "ditto" data implied by standard value factoring, and, in any case, do provide layout information and complete the rectangular shape of a table.[3] (Note that the problem of "discovering" which cells are label cells and which are value cells is straightforward because the former enclose machine-printed text and the latter are either empty or enclose handwritten text—an observation which may easily be overlooked because this problem is often one of the hardest to resolve when building automated table-understanding systems.)

Building on research, including table zoning (e.g. [11, 18, 21]) and intelligent character recognition (ICR) for machine printed text (e.g. [7, 17, 19]), we assume that we have, as input, an XML document that describes three attributes about each cell C in a table: (1) the upper-left and lower-right coordinates of C, (2) whether C is empty, and (3) the character string for C if C contains machine-printed text.[4]

[2]Intelligent character recognition for handwritten text is extremely difficult, but researchers are beginning to make progress (see for example [3, 4, 10, 20]). Their efforts, however, depend on good dictionaries, which in turn depend on knowing what kind of information an intelligent character recognizer is reading—a problem for which we provide a solution in this paper.

[3]To simplify our discussion, when we are not interested in whether an empty cell has an implied value or is simply blank, we sometimes refer to empty cells as value cells.

[4]For this paper, we do not assume that any handwritten



STATE Texas — COUNTY Lamar — TOWNSHIP OR OTHER DIVISION OF COUNTY precinct 2 (part of) Roxton — NAME OF INSTITUTION

DEPARTMENT OF COMMERCE AND LABOR—BUREAU OF THE CENSUS
THIRTEENTH CENSUS OF THE UNITED STATES: 1910—POPULATION — 30

NAME OF INCORPORATED PLACE — ENUMERATED BY ME ON THE 25th DAY OF April, 1910. B L Holloman, ENUMERATOR.

SUPERVISOR'S DISTRICT NO. 8 — ENUMERATION DISTRICT NO. 86 — SHEET NO. 11 — WARD OF CITY

	Dwelling	Family	Name	Relation	Sex	Color	Age	Marital	Yrs married	Born	Living	Birthplace	Father	Mother	Language	Trade	Industry	Class	Out of work	Weeks	Read	Write	School	Owned	Free	Farm	Schedule
1			Rivers Dee	Son	M	W	25	S				Texas	Missouri	Alabama	English	Farmer	Farm	Oa			yes	yes		R		F	163
2			— Frank	Son	M	W	21	S				Texas	Missouri	Alabama	English	Farmer	Farm	Oa			yes	yes		R		F	164
3			— Pat	Son	M	W	20	S				Texas	Missouri	Alabama	English	Laborer	Gen Farm	W			yes	yes					
4			— Carrie	Daughter	F	W	18	S				Texas	Missouri	Alabama	English	none					yes	yes	no				
5			— Carl	Son	M	W	15	S				Texas	Missouri	Alabama	English	Laborer	Home Farm	W			yes	yes	yes				
6			— Jessie	Daughter	F	W	13	S				Texas	Missouri	Alabama	English	none					yes	yes	yes				
7	177	183	Bille Silas	Head	M	B	33	M1	7			~~Virginia~~ Texas	Virginia	Virginia	English	Farmer	Farm	Oa			yes	yes		R	x	F	165
8			— Lizzie	Wife	F	B	25	M1	7			Texas	Texas	Texas	English	Laborer	Home Farm	W	no	0	yes	yes					
9			— Alice	Daughter	F	B	5	S				Texas	Virginia	Texas		none											
10			— Jude	Daughter	F	B	2	S				Texas	Virginia	Texas		none											
11	178	184	Louis Ann	Head	F	B	60	Wd		1	0	Arkansas	Virginia	Tennessee	English	Farmer	Farm	Oa			yes	yes		O	F	F	166
12	179	185	Thomas Jim	Head	M	B	62	M1	28			~~Arkansas~~	Virginia	Tennessee	English	Farmer	Farm	Oa			no	no		O	F	F	167
13			— Elizabeth	Wife	F	B	62	M1	28	0	0	Arkansas	Virginia	Tennessee	English	Laborer	H. Farm	W	no	0	yes	yes					
14	180	186	Gentry Wiley G.	Head	M	B	55	M1	35			Arkansas	Virginia	Tennessee	English	Farmer	Farm	Oa	no		yes	yes		O	F	F	168
15			— Cherry	Wife	F	B	50	M1	35	3	3	Mississippi	dont know	dont know	English	Laborer	H Farm	W	no	0	yes	no					
16			Duncan Hannah	Cousin	F	B	14	S				Texas	Texas	Texas	English	Laborer	Farm	W	no	0	no	no	yes				
17	181	187	Lee Will	Head	M	B	36	M1	15			Texas	Mississippi	Mississippi	English	Farmer	Farm	Oa			yes	yes		O	M	F	169
18			— Mary	Wife	F	Mu	34	M1	15	5	2	Texas	Mississippi	Mississippi	English	Laborer	Home Farm	W	no	0	yes	yes					
19			Burton Marion	Granddaughter	F	B	4	S				Texas	Texas	Alabama		none											
20			Duncan Joe	Hired Hand	M	B	17	S				Texas	Texas	Texas	English	Laborer	Gen Farm	W	yes	15	yes	yes	yes				
21	182	188	Rallison Pat	Head	M	B	65	Wd				Mississippi	Mississippi	South Carolina	English	Farmer	Farm	Oa			no	no		O	F	F	170
22			Johnson Henry	Head	M	Mu	23	M1	1			Texas	Texas	dont know	English	Farmer	Farm	Oa			yes	yes		R	x	F	171
23			— Sallie	Wife	F	B	25	M3	1	0	0	Texas	Arkansas	Mississippi	English	Laborer	Home Farm	W	no	0	yes	yes					
24			Duncan Jessie	Cousin	F	B	6	S				Texas	Texas	[illegible]		none											
25	183	189	Simmons John	Head	M	B	22	M1	0			Texas	dont know	Texas	English	Farmer	Farm	Oa	no		yes	yes		R	x	F	172
26			— Levy	Wife	F	Mu	17	M1	0			Texas	Tennessee	Tennessee	English	Laborer	Home Farm	W	no	0	yes	yes					
27	184	190	Rallison Alvin	Head	M	B	24	M1	0			Texas	Tennessee	Texas	English	Farmer	Farm	Oa	no	0	yes	yes		R	x	F	173
28			— Micha	Wife	F	B	22	M1	0			Texas	dont know	dont know	English	Laborer	Farm	W	no	0	yes	yes					
29	185	191	Duncan Andrew	Head	M	B	80	M1	42			Virginia	Virginia	Virginia	English	Laborer	Gen Farm	W	no	0	no	no		O	x	F	
30			— Martha	Wife	F	B	80	M1	42			Alabama	dont know	Alabama	English	none					no	no					
31			— Nora	Daughter in law	F	B	37	M1	20			Texas	dont know	dont know	English	Laborer	Gen Farm	W	yes	0	no	no					
32			— McKinley	Grandson	M	B	10	S				Texas	Texas	Texas		none						yes					
33			— Walter	Grandson	M	B	8	S				Texas	Texas	Texas		none											
34			— Charley	Grandson	M	B	3	S				Texas	Texas	Texas		none											
35	186	192	Read John	Head	M	B	40	M2	10			Texas	dont know	dont know	English	Farmer	Farm	Oa	no	0	yes	yes		R	x	F	174
36			— Daisy	Wife	F	B	27	M2	10			Texas	dont know	dont know	English	Laborer	Home Farm	W	no	0	yes	yes					
37			— Mamie	Daughter	F	B	11	S				Texas	Texas	Texas	English	Laborer	Home Farm	W	no	0	yes	yes	yes				
38			— Viola	Daughter	F	B	9	S				Texas	Texas	Texas		none											
39			— Willie	Daughter	F	B	6	S				Texas	Texas	Texas		none											
40	187	193	Lee Tom	Head	M	B	55	Wd				Texas	dont know	dont know	English	Farmer	Farm	Oa			no	no		R	x	F	175
41			— Mary	Daughter	F	Mu	21	M1	3	1	1	Texas	Texas	dont know	English	Laborer	Gen Farm	W	no	0	yes	yes					
42			— Viola	Daughter	F	Mu	17	S				Texas	Texas	dont know	English	Laborer	Home Farm	W	no	0	yes	yes					
43			— Ella	Daughter	F	B	16	S				Texas	Texas	dont know	English	Laborer	Home Farm	W	no	0	yes	yes	yes				
44			Read Arthur	Son in law	M	B	24	M1	3			Texas	South Carolina	Tennessee	English	Laborer	Gen Farm	W	no	0	yes	yes					
45			— Odie	Grandson	M	B	[illegible]	S				Texas	Texas	Texas		none											
46	188	194	Miles Antny	Head	M	B	65	Wd				Virginia	Virginia	Virginia	English	Farmer	Farm	Oa	no	0	no	no		R	x	F	176
47			— Lula	Daughter	F	B	34	Wd				Texas	Virginia	dont know	English	Laborer	Home Farm	W	no	0	yes	yes					
48			Spiggs Kate	Daughter	F	B	37	Wd				Texas	Virginia	dont know	English	Laborer	Home Farm	W	no	0	yes	yes					
49			— Kattie	Granddaughter	F	B	16	S				Texas	Texas	Texas		none					yes	yes					
50			— James	Grandson	M	B	9	S				Texas	Texas	Texas		none					yes	yes					

Figure 1: Sample Table from the 1910 U.S. Census

Given this XML input file, we exploit the geometry of the table, the readable printed text in table headings, and a given application ontology in order to produce SQL insert statements for the coordinates and types of fields for records for a relational database, which is compatible with and generated from the application ontology.

Figure 2 shows the ontology we use to illustrate our ideas and also to use as the basis for test cases we ran for our results. As Figure 2 shows, the ontology declares the concepts (boxed nodes in the graph in Figure 2, with dotted boxes representing text values and solid boxes representing object-identifier values) and the relationships among the concepts (arcs connecting boxed nodes). For example, the relationship *Father of Child* relates the concept *Father* to the concept *Child*.[5] Figure 2 also declares generalizations and specializations in *isa* hierarchies; notationally, an open triangle declares an *isa* relationship (e.g. *Child isa Person* and *Birth isa Event*. Filled-in triangles denote is-part-of hierarchies (e.g. *First Name is part of Full Name* and *Year is part of Date*. Symbols that decorate connections between relationships and concepts are cardinality constraints; specifically, they denote expected participation (e.g. a *Person* is expected to be in 1 *Family* and a *Family* is expected to consist of about 4.8 *Persons*).

Compared to previous work, our approach has both similarities and differences. We use geometry and layout, as most other researchers do (see [20] for an early journal paper on this topic, [6] and [13] for a broad survey, and [8] and [13] for an explanation of why table understanding is difficult). In addition to structure, researchers have applied a variety of techniques, including, table models [5, 9, 12, 15], grammar matching [1], and keywords along with structure [16]. The work in [9] most closely relates to our own. [9] uses frames and measures similar to ours, but we use more measures and our ontology is far richer than their frames. To the best of our knowledge, no other researchers have used images of microfilm tables as their test set.

2. RECORD RECOGNITION

In this section we describe our algorithm for generating record groups of type-labeled value cells for genealogical microfilm tables. First our algorithm independently identifies seven record-structure features resulting in seven evidence matrices.[6] We then iteratively apply six correlation rules, stopping when we either achieve convergence or we have it-

text within table cells is readable.

[5]Arcs without labels all have the default label *has* or *is for*, depending on which concept is the subject and which is the object in a relationship-naming sentence (e.g. *Person has Age* or *Age is for Person*).

[6]We identified these seven features, as well as other heuristics and parameter settings, by examining 25 tables, which we used as our development set.

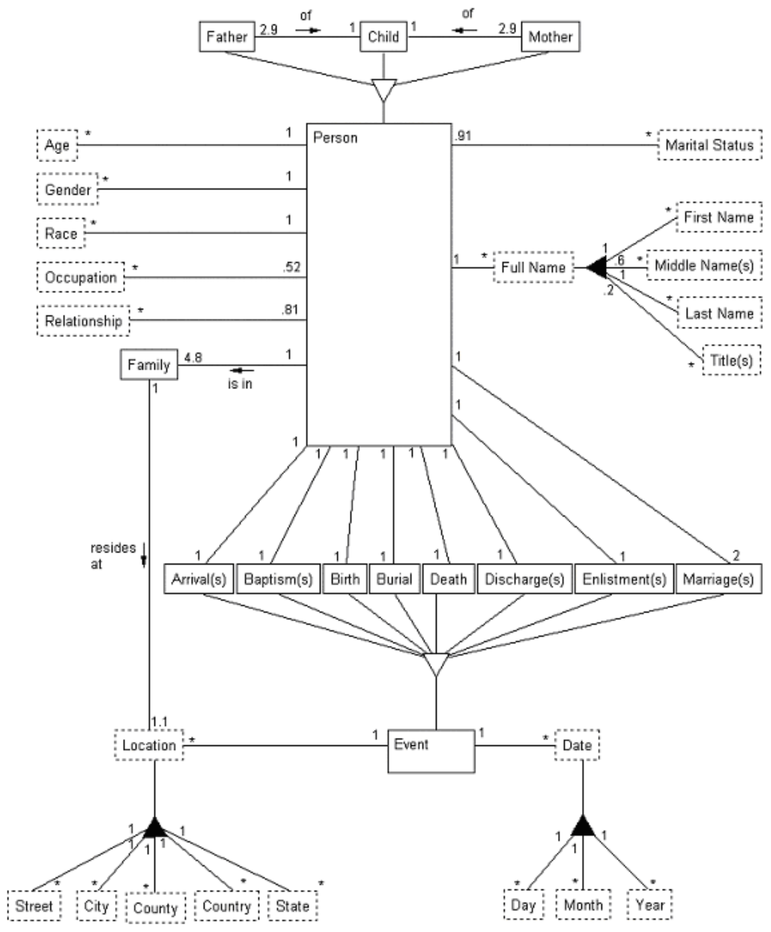


Figure 2: Genealogy Ontology—for Label Matching and Database Generation

erated more than n times, where n is a user-chosen input parameter.

2.1 Record-Structure Features

Let A be the union of the set of all label cells L, the set of all value cells V, and the set of all empty cells E. Let C be the set of all concepts in the genealogical ontology. Let $B = A \cup C$. An evidence matrix M is a $|B| \times |B|$ matrix where each value M_{ij} is a real number between 0 and 1 that represents the confidence of a relationship between elements in B. A confidence value of 0 indicates that the relationship is nonapplicable or completely uncertain, while a confidence value of 1 indicates that the relationship is completely certain. We generate confidence values by measuring features about the table cells and their relationships among themselves and with ontology concepts that support or refute the existence of the relationship expressed by M.

Feature 1. VR: Value Cells in Same Row
To determine whether value cell v_i is in the same row as value cell v_j, the algorithm uses cell coordinates to measure whether a continuous horizontal path of value or empty cells exists between v_i and v_j through a horizontal sequence of adjacent value cells or empty cells. (Note that we do not just measure the center point of v_i and v_j to see if they are within a top-to-bottom delta. Checking a sequence of horizontally adjacent cells helps us avoid skew and warp that may cause top-to-bottom deltas to yield incorrect results.) If a path exists between v_i, and v_j, the algorithm sets VR_{ij} to 1, otherwise the algorithm sets VR_{ij} to 0.[7]

Feature 2. VC: Value Cells in Same Column
Similar to **VR**, the algorithm populates the **VC** matrix by attempting to identify a continuous vertical sequence of value cells and empty cells between each value-cell pair. If a path exists between v_i and v_j, the algorithm sets VC_{ij} to 1; otherwise the algorithm sets VC_{ij} to 0.

Feature 3. VV: Value Cells Similarity
We observed from the development set that value cells in a table containing the same type of data generally have the same height and width. For each value cell pair (v_i, v_j), the algorithm multiplies the normalized difference in height and width between v_i and v_j. The normalized height (width) difference is simply the difference in height (width) divided by the largest difference in height (width) over all value cell pairs. The algorithm subtracts the product from 1 to ensure that small differences receive greater confidence values and large differences receive lesser confidence values and then stores the result in VV_{ij}.

[7]To distinguish the seven feature matrices, we name them by their two-letter code, e.g. VR for this matrix.

Feature 4. LL: Composite Labels
The algorithm considers three features, F_1, F_2, and F_3 to determine whether label cells form a composite label.[8] The algorithm computes the value of LL_{ij} for each label cell l_i and label cell l_j as the product $(F1_{ij} \times F2_{ij} \times F3_{ij})$ of the three features, $F1$, $F2$, and $F3$, which we now describe.

$F1$ Because label cells in the same composite label are likely to be close in euclidean space, $F1$ represents the distance between the midpoints of the each label cell with every other label cell. To make the $F1_{ij}$ values fall between 0 and 1, we normalize the distance by dividing the euclidean distance between the midpoints of l_i and l_j by the maximum distance between all label-cell pairs. To make small distances have high confidence values we subtract this normalized distance from 1.

$F2$ Because a composite label must eventually connect with the value cells, we determine whether a line passes through the midpoints of a pair of a label cells and through the rectangle of any value cell. The algorithm calculates the slope and intercept of the line that runs between the midpoints of a pair of label cells. It then determines whether this line passes through a value cell. If so, the pair is a potential composite label, and the algorithm sets the confidence value to be 1 for $F2_{ij}$; otherwise it sets the value to be 0.

$F3$ Because cells of a composite label should share a common boarder with at least one other cell in the composite label, $F3$ determines whether there is a common boarder. For each label cell l_i the algorithm finds all label cells that share a border to the left, right, above, and below. For example, to find a cell directly to the left of l_i, the algorithm looks at all label cells with a midpoint to the left of l_i's midpoint and between l_i's top and bottom coordinates. A label cell l_j shares a border on the left, if the distance between the left border of l_i and the right border of l_j is less than or equal to 3 pixels. (We use a threshold of 3 because the padding between label cells in the development set is no more than 3.) The algorithm calculates label cells bordering to the right, above, and below in the same way. If a border exists between l_i and l_j, the confidence value of $F3_{ij}$ is 1 and otherwise 0.

Since we compute the confidence value of LL_{ij} as $F1_{ij} \times F2_{ij} \times F3_{ij}$, it is either 0 or the value of $F1_{ij}$.

Feature 5. LV: Label Cells Describe Value Cells
A label cell at the top (left) of a column (row) typically describes the type of the value cells in the column (row). Let l_i be a label cell and v_j be a value cell. The algorithm determines if l_i heads the column or row of v_j by considering two features, $F1_{ij}$ and $F2_{ij}$, and storing their product in LV_ij.

$F1$ The first feature determines whether a continuous path of cells exists between a label cell l_i and a value cell v_j through a horizontal or vertical sequence of adjacent value cells or empty cells. If a path does not exist, the label cell does not head the row or column. The algorithm computes the path between cells by first determining the cell directly to the left, right, above, and beneath each table cell. To compute the cell directly to the left of any given cell c, the algorithm looks at all cells with a midpoint to the left of c's midpoint and between c's top and bottom coordinates. It then selects the cell with the shortest euclidean distance between its midpoint and c's midpoint. The algorithm computes the cell directly right, above, and beneath c in a similar way. By observing the development set, we discovered that genealogical tables most often contain columns of value cells headed above by a label cell. In contrast, the least common orientation occurred when a label cell described the contents of a value cell above it. To enforce this bias, we varied the confidence values assigned to label-cell/value-cell pairs in **LV** by the likelihood of their orientation. If a vertical path of cells exists between label cell l_i and value cell v_j and v_j is below (above) l_i, the $F1_{ij}$ value is 1 (0.25). If a horizontal path exists between l_i and v_j and v_j is to the right (left) of l_i, the $F1_{ij}$ is 0.75 (0.50). If a path does not exist between l_i and v_j, the $F1_{ij}$ value is 0.

$F2$ The second feature compares the height and width of each value cell with the height and width of each label cell. We observed in the development set that a value cell in a row generally has the same height as the label cell that heads the row, and a value cell in a column generally has the same width as the label cell that heads the column. To measure this, the algorithm calculates the difference in height (width) between each label-cell/value-cell pair and then normalizes the difference by dividing by the largest height (width) over all label-cell/value-cell pairs. The algorithm then subtracts this result from 1 to ensure that small differences receive high confidence values and large differences receive low confidence values.

Feature 6. LO: Label Cell Maps to Ontology
The algorithm tries to match each label cell with a concept in the ontology. As an example, printed text *The name of every person whose place of abode on the first day of June, 1870 was in this family* matches the concept *Full Name* in the ontology in Figure 2. To establish a match, the algorithm computes the product two features, $F1_{ij}$ and $F2_{ij}$ to determine whether label cell l_i matches ontology concept o_j.

$F1$ The algorithm first removes stop words (the articles, prepositions, and other common words) from the printed text associated with each label cell.[9] The algorithm then matches the synonyms[10] of the concept names in the ontology in Figure 2 with the text of the label cells, giving preference to words matched at the beginning of the character string for a label cell. Specifically, if a synonym for an ontology concept o_j matches a word in label cell l_i, the algorithm sets $F1_{ij}$ to 1 divided by the position of the matched word in the label cell's character string without the stop words. We divide

[8] Almost every label in Figure 1 is composite, i.e. consists of a general header and one or more nested subheaders.

[9] We generated the list of stop words by using the common words from the labels of the table cells in our development set.

[10] We selected synonyms by placing the ontology-concept name n in the list plus all synonyms found in labels of our set of development documents.

by this position of the word because we observed that in our development set the label cells generally have representative words at the beginning of the label's character string.

$F2$ To further weight the matched words near the beginning of a label when there are several matches, the algorithm sets $F2$ to 1 divided by the count of previously matched words in the character string of the label cell.

Feature 7. FM: Label Cell Factoring
Given label-cell/ontology-concept matches, the algorithm can use the position of an ontology-recognized label cell in relation to the position of another ontology-recognized label cell to predict whether the values of one of the label cells factors the values of the other label cell. In the development set, the value cells of a label cell l_i only factored the value cells of another label cell l_j if l_i was above or to the left of l_j. To support this preference, the algorithm determines whether the midpoint of a label cell l_i is to the left or if the midpoint of l_i is above the midpoint of label cell l_j. If so, FM_{ij} is 1 and is otherwise 0. (Note that this simply marks the possibility that the values of l_i factor the values of l_j; we need correlation-rules adjustments before we have a final answer.)

2.2 Correlation Rules

A correlation rule is an expression for updating the values of an evidence matrix using (1) the values of another evidence matrix, (2) information from the genealogical ontology, or (3) other values elsewhere in the same evidence matrix. These correlation rules attempt to find corroborating evidence to increase confidence values or to find conflicting evidence to decrease confidence values.

The algorithm iteratively applies a set of six correlation rules to refine the seven populated evidence matrices. We iterate through the rules because each rule makes only a small change to the confidence values of a matrix at each step or only performs an action when the values of in an evidence matrix reach a predefined threshold. Thus one rule cannot dominate the effects of other rules, and each rule is dependent upon the changes made by other rules. In addition, the rules work together to allow the confidence values to gradually settle. The algorithm iterates until no rules can update confidence values, or until it reaches 1000 iteration. We set the maximum number of iterations to 1000 because applying more than 1000 iterations had no effect on the results for the development set.

The first correlation rule uses the confidence values in **VV** that relate geometrically similar value cells to support the concept that value cells associated with the same label cell in **LV** should be geometrically similar. The evidence matrix **LV** stores the confidence value of a label cell heading the column or row of a particular value cell. The correlation rule disassociates a value cell that is not similar to other value cells associated with its label cell as follows. Let A be the set of all value cells that are headed by a label cell l_i with a confidence value greater than 0.5 in **LV**. For each value cell v_j in A, we consider every other value cell v_k in A (i.e. with v_j fixed, we consider all k where $1 \leq k \leq |A|$ and $j \neq k$). If the confidence value of every v_j-v_k pair is less than 0.5 in **VV**, the algorithm disassociates v_j from A by placing a 0 at position LV_{ij}. Note that i references the label cell l_i, and j references the value cell v_j to be removed.[11]

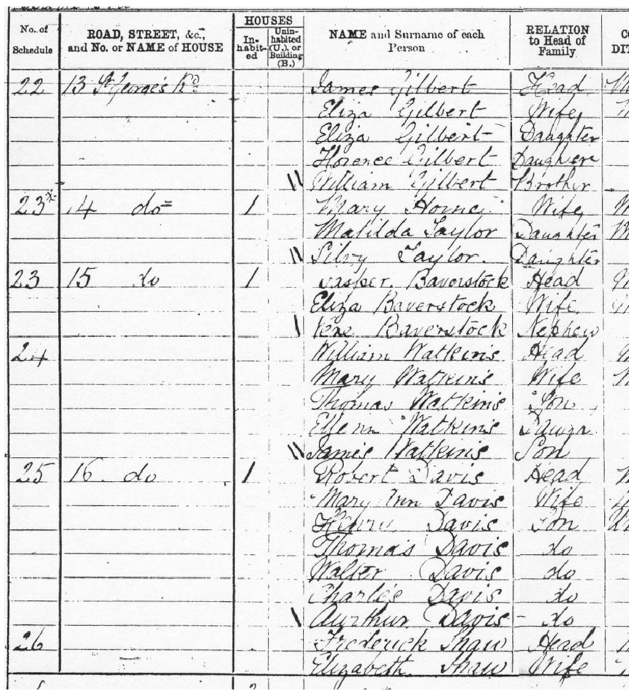
The second correlation rule reassigns a label cell that maps with high confidence to two or more subpart concepts of an is-part-of relationship in the ontology to the superpart. For example, if a label cell maps with high confidence to both *First Name* and *Last Name* in Figure 2, the correlation rule discards these two mappings and replaces them with a single high-confidence mapping from the label cell to the superpart concept *Full Name*. Specifically, the algorithm checks each label l_i that associates with two or more subpart concepts in an is-part-of relationship with a confidence value greater than 0.5 in **LO** and maps l_i to the superpart by placing a 1 in the **LO** matrix at LO_{ij} for label l_i and concept o_j and by placing 0 at LO_{ik_1} and at LO_{ik_2} for label l_i and concepts o_{k_1} and o_{k_2}.

The third correlation rule raises the confidence values for the label cells that associate with value cells in likely geometric orientations. Recall that we populated the **LV** matrix with different values depending on whether the value cell was below (1), right of (0.75), left of (0.5), or above (0.25) a label cell. The algorithm associates each label cell l_i with the all the value cells that have the highest confidence of association with l_i and increments the confidence value in the **LV** matrix 0.1 at each step of the iteration for each of these label-cell/value-cell pairs.

The fourth correlation rule distributes label-to-concept mappings across multilevel groups of label cells. We do this to combine groups of label cells into one composite label. For example, Figure 1 has a three-level group of label cells where the label cell *RELATION* maps to the concept *Relationship* in the ontology (Figure 2), the label cell *Relationship of the Person to the Head of the Family* also maps to *Relationship*, and the label cell *4* does not map to any concept in the ontology. To create a composite label, the algorithm first gathers all the multilevel groups of labels using the confidence values in the **LL** matrix. If the confidence value for two labels is greater than 0.5, then the algorithm groups the labels. We do this recursively to create groups of multilevel label cells. Next, the algorithm copies the concepts associated with each label in the group to the other concepts in the label cell in the group. In our example, the algorithm assigns *Relationship* to *4*, which is merely a column number. By distributing the concept mappings to each label cell in a composite label, we aggregate the label cells into one composite label with appropriate mappings to ontological concepts.

The fifth correlation rule refines the factoring matrix, **FM**, using the ontology **O**. Each concept has a particular expected participation in each of its incident relationships. The *4.8* in Figure 2, for example, declares that there should be approximately 4.8 people in each family. As we can see from the example in Figure 3 when we factor *NAME* by *No. of Schedule*, there are about 4.8 people in each family (by actual count there are 6 families and 25 people, which is 4.2

[11]We set the constant values in this and other correlation rules based on our observations of iteratively applying the correlation rules to the development set.



No. of Schedule	ROAD, STREET, &c., and No. or NAME of HOUSE	HOUSES In-habit-ed	HOUSES Uninhabited (U.), or Building (B.)	NAME and Surname of each Person	RELATION to Head of Family
22	13 St George's Rd			James Gilbert	Head
				Eliza Gilbert	Wife
				Eliza Gilbert	Daughter
				Florence Gilbert	Daughter
			\\	William Gilbert	Brother
23x	14 do		1	Mary Horne	Wife
				Matilda Taylor	Daughter
			\\	Silvy Taylor.	Daughter
23	15 do	1		Jasper Baverstock	Head
				Eliza Baverstock	Wife
			\	[illegible] Baverstock	Nephew
24				William Watkins	Head
				Mary Watkins	Wife
				Thomas Watkins	Son
				Ellen Watkins	Daugr
			\\	James Watkins	Son
25	16 do	1		Robert Davis	Head
				Mary Ann Davis	Wife
				[illegible] Davis	Son
				Thomas Davis	do
				Walter Davis	do
				Charles Davis	do
			\	Arthur Davis	do
26				Frederick Shaw	Head
				Elizabeth Shaw	Wife

Figure 3: Portion of a Microfilm Table with Factoring

people per family in this example). The algorithm considers each label-cell/ontology-concept pair that has a confidence value greater than 0.5 in **LO**. The algorithm updates **FM**, using the formula: $FM_{ij} = FM_{ij}(1.6 - |O_{ij} - N_j/N_i|)$, where O_{ij} is the expected participation of concept i in a relationship with concept j, N_j is the number of non-empty value cells for label cell l_j in **LV** with a confidence value greater than 0.5, N_i is the number of non-empty value cells for label cell l_i in **LV** with a confidence value greater than 0.5, and 1.6 is a parameter, which we experimentally determined using the development-set data. If the result FM_{ij} is less than 0 (greater than 1), the algorithm sets FM_{ij} to 0 (1). To illustrate, let concept i be *ROAD ...* in Figure 3 and concept j be *NAME ...*, then since *ROAD ...* is left of *NAME ...*, FM_{ij} is 1 (see Feature 7), and since $O_{ij} = 4.8$, $N_j = 25$, and $N_i = 4$, the new value for FM_{ij} is $1(1.6 - |4.8 - 25/4|) = 0.15$.[12]

The sixth correlation rule increases the confidence value for a label cell that maps to two or more concepts in the label-to-ontology-concept matrix **LO**, allowing all to be accepted when the concepts are separated by only one relationship in the ontology. When the algorithm populates **LO**, it assigns lower confidence values to concepts that match words at the end of an concept's character string. The goal of this sixth correlation rule is to increase the confidence values for these tail-end concepts if they relate closely to the other concepts that map to the label cell. Thus, if only one relationship separates two concepts, the algorithm changes the lesser of the confidence values to the greater of the two.[13] Suppose, For example, that a label cell maps to the concept *State* with a confidence value of 0.78 in **LO** and maps to the concept *Birth* with a confidence value of 0.32 in **LO**. Observe in Figure 2 that *State* is-part-of a *Location* and *Birth* is-a specialization of *Event*, and that a single relationship connects *Location* and *Event*. Hence, the algorithm increases confidence value of *Birth* to 0.78.

3. EXPERIMENTAL RESULTS

3.1 Results

We ran the algorithm on a development set of 25 tables taken from 5 different microfilm rolls and a test set of 75 tables taken from 15 different microfilm rolls (20 different microfilm rolls altogether). We measured the overall success of the algorithm by observing the accuracy of the fields in the generated SQL statements, where accuracy is the harmonic mean $2/((1/P)+(1/R))$ [2], which depends on the precision P, the number of correctly populated fields divided by the total number of populated fields, and the recall R, the number of correctly populated fields divided by the total number of value fields the algorithm should have correctly placed in the table.

We tuned the algorithm on the development set until we obtained an accuracy of 99.71%. We could not obtain 100% without some significant changes to to the way we build our factoring matrix **LO**. Figure 3 helps us see the problem.[14] The second column in Figure 3 labeled *ROAD ...* matches *Location* in our ontology (Figure 2), but the first column labeled *No. of Schedule* does not match anything in our ontology. Since we factor only with respect to columns that match our ontology, we grouped families 23 and 24 and also 25 and 26 who happen to live together in the same household as separate families.

After tuning the algorithm on the 25 development tables, we ran our algorithm on the 75 test tables. Although ground-truthing can be hard for some tables [8], we encountered no difficulty in determining correctness for the genealogical microfilm tables we encountered. Table 1 shows the results obtained by our algorithm along with a breakdown of the results for each of the seven evidence matrices after iterating over the correlation rules.

3.2 Discussion

The algorithm correctly identified all rows and columns. Indeed, it even correctly separated columns from headers and footers and rows from left and right margins. When not

[12] Note that in this example *ROAD ...* does not factor *NAME ...*; *No. of Schedule* does. Indeed, further note that if we let i be *No. of Schedule* and j be *NAME*, $FM_{ij} = 1$.

[13] We only compare two at a time, but we do compare all pairs when there are more than two label cells in a composite label.

[14] By itself, Figure 3 would have declared that there is no factoring at all (recall that we computed the factoring confidence value for *ROAD ...* factoring *NAME ...* for this page to be 0.15, too low too declare any factoring). Since microfilm rolls repeat the same filled-in table in image after image, however, we always consider several several images, not just one, when we investigate factoring. When we considered multiple images in the microfilm roll from which Figure 3 was taken, the system declared that *ROAD ...* does indeed factor *NAME ...*, i.e. in subsequent pages there were significantly fewer households comprised of more than one family.

Evidence Matrix Name	Abbreviation	Precision	Recall	Accuracy
Value-Cell Pairs (Rows)	**VR**	100	100	100
Value-Cell Pairs (Columns)	**VC**	100	100	100
Value-Cell Pairs (Similar Shape)	**VV**	100	100	100
Multilevel Labels	**LL**	100	99.67	99.82
Label-Cell / Value-Cell Pairs	**LV**	100	98.12	98.97
Label-Cell-to-Ontology Mappings	**LO**	84.98	92.76	88.18
Factored Label Cells	**FM**	100	93.40	93.47
All Database Fields		93.20	92.41	92.15

Table 1: Results for the 75 Test Tables

100% correct, its recall values for correctly composing multilevel labels, for correctly labeling value cells, for correctly mapping labels to the ontology, and for correctly recognizing the factoring in the form were all better than 90%. The precision for all evidence matrices except mapping labels to ontology concepts was 100%. Thus, except for being too aggressive in declaring label-to-ontology mappings about 15% of the time, the system never declared a false relationship in a processed evidence matrix.

Although highly accurate, it is interesting to consider the five recall and precision measures that are less than 100% in Table 1.

- *Recall for Multilevel Labels.* One of our test tables had very narrow label cells and highly skewed vertical columns. With respect to our test (Feature 4), this combination caused us to miss forming a group when it really did exist.

- *Recall for Label-Cell / Value-Cell Pairs.* In one of our test tables the label *Men Married* was the label for a double column, the first of which was for the man's last name and the second of which was for the man's first name. For this table, the algorithm mapped the label cell *Men Married* to the concept *Full Name.* Although the algorithm declares that both columns together represent a name, it was unable to separately declare the last and first names. Having access to some of the values (assuming ICR for handwriting will eventually be possible), the algorithm could be augmented to use this information and an example (which actually appeared) such as "Zwicker" in the first column and "John" in the second column to sort out first and last names.

- *Recall for Label-Cell-to-Concept Mappings.* The algorithm sometimes incorrectly matched concepts with label cells with long character strings of machine-printed text. Although not very readable, it is nevertheless clear that many of the labels in Figure 1 are long. Of the 75 test tables, 40 exhibited this problem to some degree, and the system mapped several of them correctly. As one specific incorrect example, the algorithm mapped the label cell with the machine-printed text *State here the particular Religion, or Religious Denomination, to which each person belongs. [Members of Protestant Denominations are requested not to describe themselves by the vague term 'Protestant,' but to enter the name of the Particular Church, Denomination, or Body, to which they belong.].*" Since the label cell's text begins with the word *State*, the algorithm mapped the label cell to the concept *State* for *Location* in Figure 2. Natural language processing can be used to overcome this problem by detecting the part-of-speech and meaning of these words. Since the word *State* in the label cell is a verb, it is possible to determine that the algorithm should not match it to the noun *State* that is a location.

- *Recall for Factored Label Cells.* We identified and discussed this error in connection with our explanation about why we could not obtain 100% for our development set. A potential solution to this problem would be to observe that nonempty cells, especially nonempty cells on the left, matter—in Figure 3 *No. of Schedule*, including in particular the *24* and the *26*, factor families even though the label does not match the ontology.

- *Precision for Label-Cell-to-Concept Mappings.* An interesting example happened when the system falsely declared that the label *RELIGIOUS PROFESSION* should map to the concept *Occupation.* We usually think of a profession as an occupation, but the help we needed here, again, was natural language processing, which would have identified the adjective *RELIGIOUS* as modifying the noun to give the nominalized verb a different meaning.

Finally, we observe that with an overall accuracy of 92.15%, the algorithm correctly identified records, correctly obtained each field, and correctly associated it with related fields.[15] We further note that because the tables in each microfilm roll are usually the same, it is reasonable for a human to verify the work done by the automatic record recognizer and adjust the results when they are not 100% accurate. Thus, with (possibly) minor adjustments, all records for the (approximately) 1,000 images on a roll of microfilm could be correctly identified and properly highlighted and categorized (as *Name*, *Age*, *Occupation*, *Date of Birth*, ...) for extraction work either by a human extractor, a semiautomatic extractor, or (in the future, when ICR for handwriting works well enough) a fully automatic extractor.

[15]The relational database generated from the ontology in Figure 2 included 5 tables: the first called *Person* with 21 attributes, the second called *Family* with 7 attributes, the third called *Event* with 11 attributes, the fourth called *Mother_Child* with 2 attributes, and the fifth called *Father_Child* also with 2 attributes.

4. CONCLUDING REMARKS

We have described an algorithmic process to automatically identify record patterns found in microfilm tables with machine-printed labels and handwritten values. Our approach exploited both geometric and ontological properties of tables, making use of a set of seven evidence matrices to establish initial geometric and ontological properties and then iterating over six correlation rules to corroborate evidence to strengthen or weaken the confidence values in the evidence matrices. After iterating to settle conflicts, the algorithm produced SQL insert statements for each value cell belonging to a record in the original document. The algorithm achieved a precision of 93.20 percent, a recall of 92.41 percent, and an accuracy of 92.15 percent on the database fields populated from the microfilm tables in our test set.

5. REFERENCES

[1] A. Amano, N. Asada, T. Montoymam, T. Sumiyoshi, and K. Suzuki. Table form document synthesis by grammar-based structure analysis. In *Proceedings of the Sixth International Conference on Document Analysis and Recognition (ICDAR'01)*, pages 533–537, Seattle, Washington, September 2001.

[2] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, Menlo Park, California, 1999.

[3] A. El-Nasan and G. Nagy. Ink-link. In *Proceedings of the 15th International Conference on Pattern Recognition (ICPR-2000)*, pages 573–576, Barcelona, Spain, September 2000.

[4] A. El-Nasan and G. Nagy. On-line handwriting recognition based on bigram co-occurrences. In *Proceedings of the Sixteenth International Conference on Pattern Recognition (ICPR'02)*, Quebec City, Canada, August 2002. to appear.

[5] E. Green and M. Krishnamoorthy. Model-based analysis of printed tables. In *Proceedings of the Third International Conference on Document Analysis and Recognition (ICDAR'95)*, pages 214–217, Montréal, Canada, August 1995.

[6] J. Handley. Chapter 8: Document recognition. In E. Dougherty, editor, *Electronic Imaging Technology*, pages 289–316, 1999.

[7] H. Hou. *Digital Document Processing*. Wiley, New York, New York, 1983.

[8] J. Hu, R. Kashi, D. Lopresti, G. Nagy, and G. Wilfong. Why table ground-truthing is hard. In *Proceedings of the Sixth International Conference on Document Analysis and Recognition*, pages 129–133, Seattle, Washington, September 2001.

[9] M. Hurst and S. Douglas. Layout and language: Preliminary investigations in recognizing the structure of tables. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR'97)*, pages 1043–1047, Ulm, Germany, August 1997.

[10] S. Jager. *Recovering Dynamic Information from Static, Handwritten Word Images*. PhD thesis, University of Freiburg, 1997.

[11] A. Jobbins and L. Evett. Segmenting documents using multiple lexical features. In *Proceedings of the Fifth International Conference on Document Analysis and Recognition (ICDAR'99)*, pages 721–724, Bangalore, India, September 1999.

[12] T. Kochi and T. Saitoh. User-defined template for identifying document type and extracting information from documents. In *Proceedings of the Fifth International Conference on Document Analysis and Recognition*, pages 127–130, Bangalore, Indea, September 1999.

[13] D. Lopresti and G. Nagy. Automated table processing: An (opinionated) survey. In *Proceedings of the Third IAPR Workshop on Graphics Recognition*, pages 109–134, Jaipur, India, September 1999.

[14] D. Olsen. Challenges in constructing a digital microfilm library. In *Proceedings of the Second Annual Family History Technology Workshop*, Provo, Utah, April 2002.

[15] C. Peterman, C. Chang, and H. Alam. A system for table understanding. In *Proceedings of the Symposium on Document Image Understanding Technology (SDIUT'97)*, pages 55–62, Annapolis, Maryland, April/May 1997.

[16] P. Pyreddy and W. Croft. TINTIN: A system for retrieval in text tables. In *Proceedings of the 2nd ACM International Conference on Digital Libraries*, Philadelphia, Pennsylvania, July 1997.

[17] S. Rice, G. Nagy, and T. Nartker. *Optical Character Recognition: An Illustrated Guide to the Frontier*. Kluwer Academic Publishers, Boston, Massachusetts, 1999.

[18] Y. Tang, S. Lee, and C. Suen. Automatic document processing: A survey. *Pattern Recognition*, 29(12):1931–1952, 1996.

[19] J. van Vliet. *Document Manipulation and Typography*. Cambridge University Press, Cambridge, Massachusetts, 1988.

[20] T. Watanabe, Q. Quo, and N. Sugie. Layout recognition of multi-kinds of table-form documents. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(4):432–445, 1995.

[21] K. Zuyev. Table image segmentation. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR'97)*, pages 705–708, Ulm, Germany, August 1997.