Experiments in Table Recognition

Jianying Hu[a] Ramanujan Kashi[a] Daniel Lopresti[b] Gordon Wilfong[b]

[a]Avaya Labs
Avaya Inc.
233 Mt Airy Rd,
Basking Ridge, NJ 07920, USA
{jianhu,ramanuja}@avaya.com

[b]Bell Labs
Lucent Technologies, Inc.
600 Mountain Ave,
Murray Hill, NJ 07974, USA
{dpl,gtw}@research.bell-labs.com

Abstract

Detecting and parsing tables in documents are challenging problems in document layout analysis. In this paper we report results of experiments conducted on the University of Washington database. We give detailed evaluation of the detection results using the ground-truth provided in the database and an analysis of table parsing results.

1 Introduction

Detecting and parsing tables in documents are challenging problems in document layout analysis. In earlier papers [2, 3] we addressed the problems of detecting and parsing tables in multiple media and tested our algorithms on some small databases. In this paper we report the performance of our detection and parsing algorithms on a large, standard dataset. We chose the University of Washington I CD-ROM (UW1), which contains 979 pages scanned mostly from journal articles [6, 7]. The ground-truth associated with this dataset are the zones on the document pages and the corresponding text for all the text zones. The zones are described by their bounding boxes and each zone is classified either as a text zone or a non-text zone. The non-text zones include tables, figures, half-tones etc. Our main interest in this ground-truth was specifically the non-text zones labeled as tables. However, in the case of zones marked as tables no other information about the structure or content was available. Therefore in this paper we will report a formal evaluation of the detection algorithm, but will only give an analysis of the performance of the parsing algorithm.

2 Experimental Results: Detection

The goal of the table detection algorithm is to detect the presence of one or more tables in the document. We assume that the input is a single column document zone segmentable into individual, non-overlapping text lines (referred to simply as "lines" henceforth). Our table detection algorithm does not assume any form of delimiter, rather, it computes a value for all possible starting and ending positions of tables and then choose the best possible way to partition the input document zone into zero or more number of tables. The value computed is based on the amount of white-space correlation between adjacent lines and a vertical connected component analysis [2]. For this particular experiment a zone is labeled a table zone if the algorithm found at least one table region with a high confidence score.

From the 979 pages in the entire database we used the 110 pages which contains one or more tables. All of the page images were pre-processed using Baird's **pagereader** system to identify word bounding boxes [1]. Based on the ground-truth provided in the UW1 database, there were 1368 zones (table, text, figures, etc.,) in the 110 pages chosen for this experiment. A total of 135 zones were classified as table zones in the ground-truth. The word bounding boxes for each of the 1368 zones were input to the table detection algorithm. The job of the table detection algorithm was to classify it either as a table zone or a non-table zone.

Our results indicates that at a given threshold for detection (100) 125 of the zones were correctly classified as tables and thereby missed 10 tables. An example of a missed table is seen in Figure 1. Based on the horizontal and vertical ruling lines, it seems obvious to a ground-truther that this is a table zone. However, our detection algorithm discards any ruling lines and due to the absence of good white-space correlation misses the table. Further, 17 of the zones were classified as table zones by our algorithm which were not indicated in the ground-truth. A closer analysis of these false detects pointed us to 7 zones which had an equation array. These are not tables in the true sense of the word, but do have a 2-D pattern which resembles tables. One example of this is seen in Figure 2. This zone was not classified as a table zone in the groundtruth. However, the detection al-

gorithm flagged this as a table based on a high white-space correlation score. The other 10 zones classified by our algorithm were either figures which had text which were aligned to produce a high white-space correlation score or the document had a lot of skew which resulted in incorrect segmentation. To summarize, we achieved 93% recall and 89% precision.

Table 2 Rate of forecasting heat decreases

Total results	The extent of decreasing heat level	
	1470≦T pig<1480	T pig<1470
$\frac{27\text{ taps}}{45\text{ taps}}(\times 100=60\%)$	$\frac{9\text{ taps}}{18\text{ taps}}(\times 100=50\%)$	$\frac{18\text{ taps}}{27\text{ taps}}(\times 100=67\%)$

Figure 1. A genuine table missed by the detection algorithm.

$$\alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \\ \alpha_5 \\ \alpha_6 \end{bmatrix} = \begin{bmatrix} 0.26 \\ 2.37 \\ 1.84 \\ 1.31 \\ 0.17 \\ 0.05 \end{bmatrix}.$$

Figure 2. Zone classified as table by the detection algorithm.

3 Experimental Results: Table Parsing

In our previous small scale experiments we had developed a formal method for evaluating table parsing results called *random graph probing* ([3]). Unfortunately such formal evaluation could not be performed on the UW1 database because the ground-truth there did not supply any interpretation for the tables. In attempting to provide such information ourselves, we discovered that table ground-truthing is an extremely hard problem [4]. This is partly due to the fact that UW1 is a very challenging dataset containing tables from many different domains and in vastly different layout styles. Serious research is needed before a such a large and diverse set of tables can be properly ground-truthed. For the moment, we can only provide some informal analysis of the parsing results.

Our parsing algorithm first performs column segmentation by applying hierarchical clustering to all words in the table body region to identify their likely groupings. Such groupings are represented as a binary tree constructed in a bottom-up manner [5] – first the leaf level clusters are generated where each word belongs to a unique cluster, then the two clusters with the minimum inter-cluster distance are merged into a new cluster. The merging process is repeated recursively until there is only one cluster left.

In this process each word w_i is represented by its starting and ending horizontal positions represented by the position vector $p_i = (s_i, e_i)$. The distance between two words w_i and w_j is then defined as the Euclidean distance between the two position vectors p_i and p_j. For inter-cluster distance computation, we chose to use the so called "average link". In other words, the distance between two clusters is computed as the average of the distances between all inter-cluster pairs of words.

The cluster tree generated in the above manner represents the hierarchical structure of the table body in terms of vertical grouping of words. Each cut across the tree provides one way of clustering these words. The cut where each resulting cluster corresponds to a column is found using a breadth-first traversal of the cluster tree starting from the root. A set of heuristics are applied to determine how far to go along each branch to avoid splitting (going too far) or merging (not going far enough) columns. The main heuristic currently used is that the spacing between table columns tends to be more or less even across the whole table. This approach handles imperfect vertical alignment well because of the robust nature of hierarchical clustering.

The potential column headers are then identified using a spatial/lexical distance measure and assuming typical layout rules for headers used in most tables. Hierarchical headers are represented using a tree structure. Finally row segmentation is carried out using some simple heuristics, and the first column is assumed to be the row header column if no column header has been detected for the first column. Currently no detection of potential hierarchical row headings is carried out. A detailed description of the algorithm along with a formal evaluation of the algorithm on a set of ASCII tables can be found in an earlier paper [3].

The parsing algorithm was applied to the 125 zones correctly detected as tables. Out of these 101 received completely correct column segmentation, which is impressive considering the high level of difficulty of the dataset. Figure 3 shows one of the tables (from document `e02j`) segmented correctly. Notice that the column margins in this table are very small and the layout is somewhat irregular, making it a challenging case. The majority of the segmenta-

tion errors occurred on tables with very low scanning quality or a high level of irregularity (e.g., large number of rows spanning multiple columns).

TABLE 1. *Analysis of covariance for gas exchange parameters A, through sustained HL; B, comparison of morning vs. afternoon with intervening HL period*

Effect	P_{net} F	P_{net} P	g_s F	g_s P	C_i F	C_i P
A. Sustained HL						
12 June						
Time	4.93	*[a]	21.34	***	17.85	***
Plant	2.55		7.15	***	4.34	**
Plant × log(PPF)	2.47		7.13	***	4.30	**
C_a					219.93	***
R^2	0.75		0.84		0.96	
17 June						
Time	4.32	*	26.88	***	0.10	
Plant	2.01		4.83	**	0.47	
Plant × log(PPF)	1.75	**	4.85	**	0.40	
C_a					1.14	
R^2	0.83		0.88		0.34	
19 June						
Time	9.22	***	47.71	***	1.58	
Plant	4.55	**	0.95		2.46	
Plant × log(PPF)	4.53	**	0.89		2.43	
C_a					35.55	***
R^2	0.72		0.84		0.61	
B. Morning vs. afternoon						
5 June						
Time	40.96	***	238.49	***	1.54	
Plant	3.80	*	0.59		8.16	**
Plant × log(PPF)	3.81	*	0.39		7.51	**
C_a					61.78	***
R^2	0.92		0.98		0.93	
7 June						
Time	382.50	***	210.24	***	82.30	***
Plant	55.20	***	5.41	*	5.02	
Plant × log(PPF)	55.15	***	5.55	*	4.88	
C_a					21.11	**
R^2	0.98		0.97		0.94	
17 June						
Time	69.29	***	186.23	***	59.91	***
Plant	3.08	*	20.66	***	16.38	**
Plant × log(PPF)	2.74	*	24.18	***	17.80	***
C_a					28.08	***
R^2	0.95		0.96		0.95	
19 June						
Time	69.31	***	164.14	***	48.62	***
Plant	136.88	***	11.56	***	68.88	***
Plant × log(PPF)	152.30	***	10.64	**	71.10	***
C_a					182.53	***
R^2	0.99		0.98		0.99	

[a] Significance levels: * $P < 0.05$, ** $P < 0.01$, *** $P < 0.001$.

Figure 3. A sample table and its column segmentation.

Evaluating row segmentation and header identification turned out to be more difficult on this dataset. Row segmentation is non-trivial because there are often multi-line rows with no clear boundaries in between. In our attempt to provide detailed ground truth, we found that many of the 135 tables were somewhat anomalous and difficult to interpret even for humans, particularly when one tried to identify the rows and headers. By examining the results by hand, we found that for 63 of the tables we could say with confidence that they received completely correct row segmentation, and only 26 received completely correct header detection. Again, many errors were caused by high levels of noise or irregularity. In particular, many row segmentation errors stemmed from line segmentation errors made by the lower level page segmentation program due to noise or large skew. Another major group of tables where our row segmentation failed are tables with embedded hierarchical row headings such as the one shown in Fig. 3. In this case there are three levels of row headings; the highest level including "A. Sustained HL" and "B. Morning vs. afternoon"; the second level including "12 June", "17 June", etc; and the lowest level including "Time", "Plant", etc. All three levels are laid out in one column and there is little spatial cue indicating the hierarchical structure. For example, without semantic analysis there is little evidence that the row containing "12 June" should not be grouped with the row below, making "12 June" and "Time" a multi-line header. Similar analysis applies to header detection. In both cases our current algorithm relies mostly on spatial cues to infer the structure, which is not always adequate. Sophisticated language analysis is clearly needed in order to handle the more general cases. Finally, the current algorithm does not make use of the occasional demarcation lines. Use of such lines would also improve parsing results.

4 Conclusion

This paper has presented experiments in table detection and table parsing using the UW1 database. Due to the nature of the ground truth available the table detection problem was tested as a zone classification problem and our algorithm achieved 93% recall and 89% precision. We intend to expand the scope of this experiment to include all the documents in the UW1 database. No formal evaluation was carried out for table parsing due to lack of ground truth. However, an analysis of the parsing results indicated reasonable performance in terms of "physical structure" identification, in particular, column segmentation. It is also clear from such analysis that natural language processing is the key to improving the performance of table parsing, particularly in handling generic tables from multiple domains.

References

[1] H. Baird. Anatomy of a versatile page reader. *Proceedings of the IEEE*, 80(7):1059–1065, 1992.

[2] J. Hu, R. Kashi, D. Lopresti, and G. Wilfong. Medium-independent table detection. In *Proceedings of Document Recognition and Retrieval VII (IS&T/SPIE Electronic Imaging)*, pages 291–302, San Jose, CA, January 2000.

[3] J. Hu, R. Kashi, D. Lopresti, and G. Wilfong. Table structure recognition and its evaluation. In *Proceedings*

of Document Recognition and Retrieval VIII, volume 4307, pages 44–55, San Jose, CA, January 2001.

[4] J. Hu, R. Kashi, D. Lopresti, G. Wilfong, and G. Nagi. Why table ground-truthing is hard. In *to be presented at ICDAR 2001*, Seattle, WA, September 2001.

[5] A. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.

[6] I. Phillips, S. Chen, and R. Haralick. CD-ROM document database standard. In *Proceedings of Second International Conference on Document Analysis and Recognition*, pages 478–483, Tsukuba Science City, Japan, October 1993.

[7] I. Phillips, J. Ha, R. Haralick, and D. Dori. The implementation methodology for the CD-ROM English document database. In *Proceedings of Second International Conference on Document Analysis and Recognition*, pages 484–487, Tsukuba Science City, Japan, October 1993.