2013 12th International Conference on Document Analysis and Recognition

ICDAR 2013 Table Competition

Max Göbel*, Tamir Hassan†, Ermelinda Oro‡ and Giorgio Orsi§
*Technische Universität Wien, Austria. Email: mcgoebel@gmail.com
†Universität Konstanz, Germany. Email: tamir.hassan@uni-konstanz.de
‡ICAR-CNR, Università della Calabria, Italy. Email: oro@icar.cnr.it
§Dept. of Computer Science, University of Oxford, UK. Email: giorgio.orsi@cs.ox.ac.uk

***Abstract*—Table understanding is a well studied problem in document analysis, and many academic and commercial approaches have been developed to recognize tables in several document formats, including plain text, scanned page images and born-digital, object-based formats such as PDF. Despite the abundance of these techniques, an objective comparison of their performance is still missing. The Table Competition held in the context of ICDAR 2013 is our first attempt at objectively evaluating these techniques against each other in a standardized way, across several input formats. The competition independently addresses three problems: (i) table location, (ii) table structure recognition, and (iii) these two tasks combined. We received results from seven academic systems, which we have also compared against four commercial products. This paper presents our findings.**

I. Introduction

The problem of analysing tables in documents has been dealt with in many academic publications over the previous two decades. The systems reported in the literature vary according to the input document format (ASCII text, HTML, image, PDF) and the type of structure that is recognized and output by the system.

Table understanding has been gaining traction since the beginning of the big data era due to the massive amounts of tabular data in documents on the Web. In addition, private, public and governmental institutions often publish reports in PDF format. Such data is immensely valuable for decision support and object search, but without reliable table understanding techniques, it cannot be easily indexed by search engines or used by automatic data processing applications.

In our previous work [1], we proposed a methodology for evaluating these approaches independently from the format of the input and of the output. In particular, the problem of table understanding was split up into three tasks:

1) *table location:* locating the regions of a document with tabular content;
2) *table structure recognition:* reconstructing the cellular structure of a table;
3) *table interpretation:* recovering the meaning of the tabular structure; this includes:
   a) *functional analysis:* determining the function of cells and their abstract logical relationships;
   b) *semantic interpretation:* understanding the semantics of the table in terms of the entities represented in the table, their attributes with corresponding values, and the mutual relationships between such entities.

Due to the relatively small number of approaches for table interpretation, we decided to focus on the first two tasks, namely table location and table structure recognition, in this competition.

Many approaches to table understanding, e.g. [2], [3], have been designed to work on object-based documents as input and therefore cannot be evaluated using datasets consisting solely of page images. By choosing born-digital PDF as the format for the competition dataset, we have made it possible for such approaches to participate in the competition, as well as those based on raster images and plain text documents. The methods used for performance evaluation (see Section III) are based on [1], and this competition is the first large-scale test of these metrics. Thus, not only the participating systems have been evaluated in this competition, but the metrics as well.

The results show that the best performing systems deliver an average accuracy in the 84% to 87% range for the complete process. This is not surprising given the objective difficulty of these tasks, but it also shows that we are still far from the level of accuracy that is necessary to reliably use these methods to automatically process tabular data. On average, commercial systems perform better than academic systems, but there are cases where academic systems still show a considerable advantage in addressing specific recognition issues.

II. The competition

The main objective of this competition was to obtain an overview of current methods for table detection and understanding, developed for a variety of input formats, both from academia and the commercial marketplace, and to evaluate their strengths and weaknesses. A secondary objective was to evaluate the evaluation strategy proposed in [1], and this competition presents the first large-scale test of these methods. Furthermore, we hope that the ground-truthed dataset that we have generated for this competition will prove useful to researchers in table recognition well beyond ICDAR 2013.

The dataset[1] referenced in [1] served as the official practice dataset for the competition. Rather than concentrate on one particular sub-class of documents, it has always been our intention to evaluate systems as generically as possible, and this dataset, as well as the actual competition dataset, were generated by systematically collecting PDFs from a Google

[1]downloadable from `http://www.tamirhassan.com/dataset/`

1520-5363/13 $26.00 © 2013 IEEE
DOI 10.1109/ICDAR.2013.292

search in order to make the selection as objective as possible. In order to obtain documents whose publications are known to be in the public domain, we limited ourselves to two governmental sources with the additional search terms `site:europa.eu` and `site:*.gov`.

From the collected documents, we searched for tables that meet our criteria (unambiguous bounding box and cell structure; no super- or subscript) and extracted *excerpts* containing approximately two pages before and after the table, in order to give the algorithms ample opportunity to find false positives. For the practice dataset, we included 59 excerpts with a total of 117 tables. For the competition dataset, we found a further 77 excerpts with a total of 156 tables.

As we wanted to evaluate the systems in a generic way, we kept the size of the practice dataset small to discourage attempts at training systems to a particular document class or source. Rather, the practice dataset was provided to enable participants to modify their algorithms in good time to accept PDF as input and return results in our specified XML format, as well as correct any bugs that only became apparent when running on our dataset. However, we did not specifically disallow training and note that many of the submissions have relied on it to a greater or lesser extent.

We also released a number of tools to enable the participants to automatically compare their result to our ground truth (GT), visualize their results and even make adjustments by scaling or inverting coordinates to ensure that the output corresponds to the PDF coordinate system.

The competition was run in off-line mode. Two weeks before the submission deadline, the competition dataset, without ground truth, was made available to the participants. It should be noted that there is a large degree of trust that the results submitted by the authors are genuine and that no training on the competition dataset took place. The organizers have faith in all participants' scientific integrity.

The competition was originally split up into the first two sub-competitions described below, with participants having the choice to participate in either one, or both sub-competitions:

A. Table location sub-competition (LOC)

The aim of this sub-competition was to evaluate how good the methods are at locating tabular regions on a page. (There were no tables spanning across several regions or pages in the dataset.) Participants were asked to return the bounding boxes (`x1`, `y1`, `x2`, `y2`) and the page number of each found region. For this sub-competition, we received eight valid entries, including two variations of a single algorithm, and we also obtained results from four commercial systems.

B. Table structure detection sub-competition (STR)

The aim of this sub-competition was to evaluate the structure recognition process in isolation and we therefore asked participants to return the result of their algorithm given correct (manually specified or corrected) information about each region on the page. For each cell, participants were required to return the textual content, as well as column number attributes (`start-col`, `start-row` and, for spanning cols/rows, `end-col` and `end-row`). The performance evaluation metric (see Section III) did not require any coordinate information. For this sub-competition, we only received three valid entries.

C. Comparison against commercial approaches (COM)

After we began analysing commercial products for comparison with the participants' algorithms, it became clear that several of the products did not allow the table location result to be corrected before recognizing the table structure. Furthermore, one participant was also not able to modify her system in time and was therefore also only able to submit table structure results for the "complete process". Therefore, we decided to generate a further set of results for table structure recognition based on the system's table location result. With the four commercial systems, we were able to compare seven different approaches in this result set. We are particularly grateful to all participants who submitted a result to this third "sub-competition" at such short notice.

III. Performance evaluation

The performance evaluation was carried out based on our evaluation strategy as proposed in [1], which is summarized in the following two sub-sections.

A. Comparing region results (LOC)

The measures *completeness* and *purity*, which are well defined in the context of page segmentation, were first introduced in [4]. Broadly speaking, a region is classified as *complete* if it includes all sub-objects in the GT region; a region is classified as *pure* if it does not include any sub-objects which are not also in the GT region. A correctly detected region is therefore both complete and pure.

Our original evaluation strategy was to calculate the number of complete and pure tables over the whole dataset. However, it became apparent that these measures do not discriminate between minor errors (e.g. part of a heading missing) and major errors (e.g. large parts of a region missing). This is why we additionally calculated precision and recall measures on the sub-object level in each region.

We defined the sub-objects to be the individual characters in the PDF, as this was determined to be an unambiguous, readily available feature closely related to the PDF itself. This has also the advantage that it returns exactly the same results, regardless of how tightly the region boundaries have been drawn, as only the text objects within the region play a role in the calculation.

B. Comparing structure recognition results (STR, COM)

We compare two table structures by generating a list of all *adjacency relations* between each content cell and its nearest horizontal and vertical neighbours. No adjacency relations are generated between blank cells or a blank cell and a content cell. An adjacency relation is a tuple containing the textual content of both cells, the direction and the number of blank cells (if any) in between. This 1-D list of adjacency relations can be compared to the ground truth by using precision and recall measures.

This method provides a simple, repeatable way to fairly account for a wide variety of errors in table structure recognition (e.g. extra blank columns, split rows, undetected colspans, etc.) As no coordinate information is used, result files in HTML, text and other formats can also be easily evaluated using this method. In order to account for possible character encoding issues, each content string was normalized by removing whitespace, replacing all special characters with an underscore and converting all lowercase letters to uppercase.

C. Alternative ground truths

Although great care was taken in avoiding excerpts containing ambiguous tables when generating the dataset, some of these ambiguities only became apparent when analysing the participants' submissions. Therefore, "alternative" ground truth files were later generated for four of the excerpts in the dataset. Where there were discrepancies between the ground truths in generating the numerical results, the ground truth returning the better numerical result was always chosen.

D. Combining results

There are several ways to average the precision and recall scores over the complete dataset. For both region and structure results, we chose to first calculate these scores for each document separately and then calculate the average based on the document scores. This way, each document has equal weighting and the result is not skewed by the few documents containing tables with hundreds or thousands of cells.

Because of the relatively small number of tables in a single document, we chose not to do this for completeness and purity and simply totalled the number of complete and pure tables over the complete dataset.

IV. Participating methods

The following subsections describe the various systems that have participated in the competition. A summary of the main features is given in Table I.

Participant	Format	Internal model	Methodology	Sub-competitions
Fang et al.	PDF	Objects	Heuristics	LOC
Nurminen	PDF	Img. & obj.	Heuristics	LOC, STR, COM
Yildiz	PDF	Text lines	Heuristics	LOC, COM
Silva	TXT	Text lines	Heur. + ML	LOC, STR, COM
Stoffel	PDF	Text lines	Heur. + ML	LOC
Hsu et al.	Images	Objects	Heuristics	LOC, STR
Liu et al.	PDF	Objects	Heuristics	LOC

TABLE I
SUMMARY OF THE MAIN FEATURES OF EACH PARTICIPATING METHOD

A. ICST-Table system, Fang et al.

The ICST-Table system [5] was submitted by Jing Fang, Leipeng Hao, Liangcai Gao, Xin Tao and Zhi Tang from the Institute of Computer Science & Technology, Peking University, Beijing, China and is designed to recognize tables in born-digital PDFs, which are parsed using a commercial library. The heuristic approach locates tables by finding whitespace and line separators and filtering out regions containing paragraphs of text. It is worth noting that in [5] authors compared their evaluation results with those presented by Liu et al. in [6], obtaining better precision and recall. In this competition, we were able to compare the two systems directly, and this time Liu et al. obtained better results on our dataset.

B. Tabler system, Nurminen

Anssi Nurminen developed the *Tabler* system as part of his MSc degree at Tampere University of Technology, Finland. The system processes born-digital PDF documents using the Poppler library and combines raster image processing techniques with heuristics working on object-based text information obtained from Poppler in a series of processing steps.

C. pdf2table system, Yildiz

Burcu Yildiz developed the *pdf2table* system [7] at the Information Engineering Group, Technische Universität Wien, Austria. The system employs several heuristics to recognize tables in PDF files having a single column layout. For multi-column documents, the user can specify the number of columns in the document via a user interface; however, such user input was not allowed in the competition. The approach was able to handle most of the documents where the tables span the entire width of the page. However, the issue of false positives was not properly addressed, as in the original workflow these would have been discarded via user interaction.

D. TABFIND algorithm, Silva

Ana Costa e Silva, from the Laboratory of Artificial Intelligence and Decision Support (LIAAD-INESC), Porto, Portugal, used an algorithm that works on textual files line-by-line, and the PDF dataset was therefore converted into text format, resulting in loss of information. The method used in the competition differs somewhat from the one presented in her thesis [8] and was adapted specifically for the competition dataset by assuming, for example, that tables have at least one line where all cells are non-empty. Furthermore, the algorithm also incorporates a training procedure for parameter tuning.

E. Stoffel's system

Andreas Stoffel, from the Department of Computer and Information Science, University of Konstanz, Germany, participated with a trainable system [9], [10] for the analysis of PDF documents based on the PDFBox library. After initial column and reading-order detection, logical classification is performed on the line level. In order to detect tables, the system was trained on the practice dataset using a sequence of a decision-tree classifier and a conditional random field (CRF) classifier. Consecutive lines labelled as tabular content were then grouped together and output as a table region.

F. KYTHE system, Hsu et al.

The *Kansas Yielding Template Heuristic Extractor (KYTHE)* was submitted by William H. Hsu (group leader), Xinghuang Xu and Jake Ehrlich from the Department of Computing and Information Sciences, Kansas State University, in collaboration with Praveen Koduru of iQGateway LLC.

KYTHE is designed to process scanned documents by using an OCR tool such as Tesseract. The approach combines automatic preprocessing (using lists of expected attributes and template-based constraints) with interactive postprocessing, enabling the system to be adapted for a specific data source.

For the competition, the PDF documents were first rasterized into bitmaps, resulting in information loss. Combined with the additional error rate of the OCR process, we can see why this approach did not perform as well as those working directly on the PDF object or text levels.

G. PSU-TableSeer system, Liu et al.

The *TableSeer* system [6] was developed by Ying Liu, Kun Bai, Prasenjit Mitra and C. Lee Giles at The College of Information Sciences and Technology, Pennsylvania State University. The submission was prepared by Sagnik Ray Choudhury and Hung-Hsuan Chen, who submitted two result sets obtained by different versions of the algorithm. The second version relaxed some conditions, increasing recall with only a slight cost to precision when tested on the competition dataset. The algorithm uses a heuristic approach by first joining together adjacent text lines with uniform font size, before using whitespace and textual cues to determine which blocks contain a table. As many of these rules are specific to research papers, the system did not perform particularly well on the competition dataset.

H. Commercial systems

We also included four commercial, off-the-shelf systems in our comparison. For each system, we generated a set of results for table location (LOC) and for table structure recognition based on the table location result (COM):

1) **ABBYY FineReader 11.0 Corporate Edition:** Each document was loaded, automatically analysed and saved as as HTML. The options to split facing pages and automatically rotate pages were disabled, as these were found to cause problems with certain documents. The region result file was generated using our interactive GT tool. We used a script to automatically convert the HTML table structure to our competition format.
2) **Adobe Acrobat XI Pro:** Each document was loaded and saved as HTML, which automatically ran Acrobat's analysis procedure. Despite trying several options, the save process failed on two of the input documents and a further document produced an empty file. As with FineReader, the same script was used to convert the HTML table structures to our competition format. The region result file was manually generated based on the content of the result tables.
3) **OmniPage 18 Professional:** Each document was loaded, automatically analysed and saved in OmniPage's proprietary XML format, which represents table structure information in a similar way to our competition format. We used a script to convert this file to our competition format. The region result file was manually generated based on the content of the result tables.
4) **Nitro Pro 8:** The "To Excel" conversion function of Nitro outputs all detected tables in Excel format (one file per document; one worksheet per page). We used our interactive GT generation tool to manually generate result files for the cell structure as well as the region structure.

V. Discussion

The result tables summarizing each of the three result sets are given in Tables II–IV and show that, on average, the best systems are able to deliver an accuracy between 84% and 87% for the complete process (see Table IV). This is still far from ideal, especially if the output of such systems is intended for fully automatic processing or analysis. A 15% error rate means that we require human verification before processing the data. It is also worth noting that completeness and purity did not always relate to the F_1-measure; both versions of Liu et al.'s system achieved a higher F_1-score than Fang, even though they did not manage to detect a single table completely.

In general, the commercial systems seem to be superior to the academic systems (only Nurminen achieved comparable performance in both sub-competitions), but they also appear to rely more on the presence of ruling lines than academic systems. As the details of the algorithms used by commercial systems are not publicly available, it is not clear whether their advantage resides in a better approach to the problem or in the fact that, over the years, a large number of ad-hoc heuristics have been added to deal with a wide variety of special cases. On the other hand, the academic participants had the opportunity to test their systems on a practice dataset for bug fixing or training, although it is not clear whether all participants made use of this opportunity.

The US dataset was found to be much more difficult than the EU dataset, especially due to a higher frequency of non-ruled tables and complex header structures, which caused problems for most of the algorithms, whereas spanning rows and columns in the body did not. To further investigate this observation, we compared the performance of each system for ruled and unruled tables separately (Figure 1). Apart from Yildiz and Silva, all systems fared better with ruled tables.

Very small tables, with fewer than five rows, also frequently caused difficulties. Many approaches find tables by growing an initial seed candidate outwards. Small tables often remained undetected by such systems.

VI. Critique

This was the first time that this competition had been run, and this section presents our experiences and suggestions for future improvement.

Early on in the process, a large number of initial submissions contained errors, which led to delays while they were corrected. A number of tools were made available to enable participants to verify their submission, but not all participants made use of this opportunity. Further development of these tools, and the inclusion of a "pre-flight" check as part of the submission process, should reduce the number of invalid submissions in the future.

	Per-document averages			Tables found (total=156)	
Participant	Recall	Precision	F_1-meas.	Complete	Pure
FineReader	*0.9971*	*0.9729*	***0.9848***	*142*	*148*
OmniPage	*0.9644*	*0.9569*	***0.9606***	*141*	*130*
Silva	0.9831	0.9292	**0.9554**	149	137
Nitro	*0.9323*	*0.9397*	***0.9360***	*124*	*144*
Nurminen	0.9077	0.9210	**0.9143**	114	151
Acrobat	*0.8738*	*0.9365*	***0.9040***	*110*	*141*
Yildiz	0.8530	0.6399	**0.7313**	100	94
Stoffel	0.6991	0.7536	**0.7253**	79	66
Liu et al. 2	0.3355	0.8836	**0.4864**	0	29
Hsu et al.	0.4601	0.3666	**0.4080**	39	95
Fang et al.	0.2697	0.7496	**0.3967**	28	41
Liu et al. 1	0.2207	0.8885	**0.3536**	0	25

TABLE II
RESULTS FOR THE TABLE LOCATION (LOC) SUB-COMPETITION

	Per-document averages		
Participant	Recall	Precision	F_1-measure
Nurminen	0.9409	0.9512	**0.9460**
Silva	0.6401	0.6144	**0.6270**
Hsu et al.	0.4811	0.5704	**0.5220**

TABLE III
RESULTS FOR THE TABLE STRUCTURE RECOGNITION (STR) SUB-COMPETITION (BASED ON CORRECT REGION INFORMATION)

	Per-document averages		
Participant	Recall	Precision	F_1-measure
FineReader	*0.8835*	*0.8710*	***0.8772***
OmniPage	*0.8380*	*0.8460*	***0.8420***
Nurminen	0.8078	0.8693	**0.8374**
Acrobat	*0.7262*	*0.8159*	***0.7685***
Nitro	*0.6793*	*0.8459*	***0.7535***
Silva	0.7052	0.6874	**0.6962**
Yildiz	0.5951	0.5752	**0.5850**

TABLE IV
TABLE STRUCTURE RECOGNITION RESULTS FOR THE COMPLETE PROCESS (COM) - BASED ON THE SYSTEM'S TABLE LOCATION RESULT

Our evaluation metrics were found to be a fair representation of the actual quality of the output from the various systems. The combination of completeness and purity with precision and recall on the character level gives a good overall picture of the region detection quality. Similarly, we have found that using cell adjacency relations to evaluate table structure detection enables us to obtain precision and recall measures which are repeatable and accurately reflect the quality of the result.

By calculating the results for each document first, we were able to reduce the bias of "data-heavy" tables on the overall result. A further improvement for the future would be to evaluate regions by calculating the area (in square points) of region overlap instead of counting characters, after "normalizing" each region first by shrinking it to the smallest region encompassing all characters within its bounds. This would avoid regions containing overprinted or non-printing characters skewing the result.

The structure results for the complete process (see Table IV) should also be treated with some caution. A number of systems

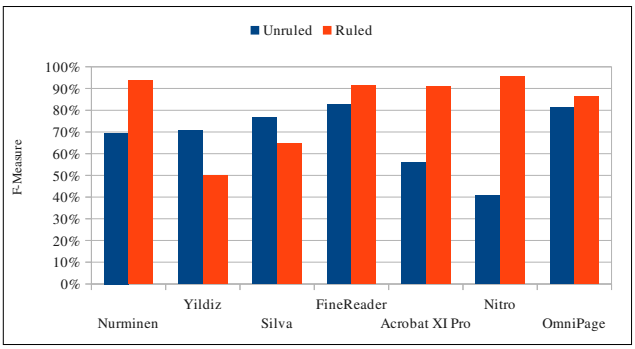

Fig. 1. Comparison of results with ruled versus unruled tables for the complete process sub-competition

returned large false positive regions, whose table structure consisted of only one cell. In many cases, this huge cell only neighboured one or two other cells, and therefore did not raise the overall false positive count significantly.

A further issue with our structure recognition metric is in the comparison of adjacency relations by their textual content. Although our normalization routine stripped or replaced most special characters, there were still some remaining encoding issues when evaluating certain approaches. This is a double-edged sword, as removing all non-alphanumeric characters would make it no longer possible to distinguish between cells that do not contain at least one letter or number, of which there were many in our dataset. In the future, we will therefore consider requiring further information about the cell, such as a bounding box, to enable its unique identification.

ACKNOWLEDGMENTS

This work has been supported by the EU FP7 Marie Curie Zukunftskolleg Incoming Fellowship Programme, University of Konstanz (grant no. 291784), the ERC grant agreement DIADEM (no. 246858) and by the Oxford Martin School (grant no. LC0910-019).

REFERENCES

[1] M. C. Göbel, T. Hassan, E. Oro, and G. Orsi, "A methodology for evaluating algorithms for table understanding in PDF documents," in *ACM Symposium on Document Engineering*, 2012, pp. 45–48.
[2] E. Oro and M. Ruffolo, "PDF-TREX: An approach for recognizing and extracting tables from PDF documents," in *Proc. of ICDAR*, 2009, pp. 906–910.
[3] B. Krüpl and M. Herzog, "Visually guided bottom-up table detection and segmentation in web documents," in *WWW*, 2006, pp. 933–934.
[4] A. C. e Silva, "Metrics for evaluating performance in document analysis: application to tables," *IJDAR*, vol. 14, no. 1, pp. 101–109, 2011.
[5] J. Fang, L. Gao, K. Bai, R. Qiu, X. Tao, and Z. Tang, "A table detection method for multipage PDF documents via visual seperators and tabular structures," in *ICDAR*, 2011, pp. 779–783.
[6] Y. Liu, K. Bai, P. Mitra, and C. L. Giles, "TableSeer: automatic table metadata extraction and searching in digital libraries," in *JCDL*, 2007, pp. 91–100.
[7] B. Yildiz, K. Kaiser, and S. Miksch, "pdf2table: A method to extract table information from pdf files," in *IICAI*, 2005, pp. 1773–1785.
[8] A. C. e Silva, "Parts that add up to a whole: a framework for the analysis of tables," Ph.D. dissertation, The University of Edinburgh, 2010.
[9] H. Strobelt, D. Oelke, C. Rohrdantz, A. Stoffel, D. A. Keim, and O. Deussen, "Document cards: A top trumps visualization for documents," *IEEE Trans. Vis. Comput. Graph.*, vol. 15, no. 6, pp. 1145–1152, 2009.
[10] A. Stoffel, D. Spretke, H. Kinnemann, and D. A. Keim, "Enhancing document structure analysis using visual analytics," in *SAC*, 2010, pp. 8–12.