Image-based table recognition: data, model, and evaluation

Xu Zhong
IBM Research Australia
60 City Road, Southbank
VIC 3006, Australia
peter.zhong@au1.ibm.com

Elaheh ShafieiBavani
IBM Research Australia
60 City Road, Southbank
VIC 3006, Australia
elaheh.shafieibavani@ibm.com

Antonio Jimeno Yepes
IBM Research Australia
60 City Road, Southbank
VIC 3006, Australia
antonio.jimeno@au1.ibm.com

***Abstract*—Important information that relates to a specific topic in a document is often organized in tabular format to assist readers with information retrieval and comparison, which may be difficult to provide in natural language. However, tabular data in unstructured digital documents, *e.g*. Portable Document Format (PDF) and images, are difficult to parse into structured machine-readable format, due to complexity and diversity in their structure and style. To facilitate image-based table recognition with deep learning, we develop the largest publicly available table recognition dataset PubTabNet[1], containing 568k table images with corresponding structured HTML representation. PubTabNet is automatically generated by matching the XML and PDF representations of the scientific articles in PubMed Central™ Open Access Subset (PMCOA). We also propose a novel attention-based encoder-dual-decoder (EDD) architecture that converts images of tables into HTML code. The model has a structure decoder which reconstructs the table structure and helps the cell decoder to recognize cell content. In addition, we propose a new Tree-Edit-Distance-based Similarity (TEDS) metric for table recognition. The experiments demonstrate that the EDD model can accurately recognize complex tables solely relying on the image representation, outperforming the state-of-the-art by 9.7% absolute TEDS score.**

I. Introduction

Information in tabular format is prevalent in all sorts of documents. Compared to natural language, tables provide a way to summarize large quantities of data in a more compact and structured format. Tables provide as well a format to assist readers with finding and comparing information. An example of the relevance of tabular information in the biomedical domain is in the curation of genetic databases in which just between 2% to 8% of the information was available in the narrative part of the article compared to the information available in tables or files in tabular format [1].

Tables in documents are typically formatted for human understanding, and humans are generally adept at parsing table structure, identifying table headers, and interpreting relations between table cells. However, it is challenging for a machine to understand tabular data in unstructured formats (*e.g*. PDF, images) due to the large variability in their layout and style. The key step of table understanding is to represent the unstructured tables in a machine-readable format, where the structure of the table and the content within each cell are encoded according to a pre-defined standard. This is often referred as *table recognition* [2].

This paper solves the following three problems in image-based table recognition, where the structured representations of tables are reconstructed solely from image input:

a) Data: We provide a large-scale dataset PubTabNet, which consists of over 568k images of heterogeneous tables extracted from the scientific articles (in PDF format) contained in PMCOA. By matching the metadata of the PDFs with the associated structured representation (provide by PMCOA[2] in XML format), we automatically annotate each table image with information about both the structure of the table and the text within each cell (in HTML format).

b) Model: We develop a novel attention-based encoder-dual-decoder (EDD) architecture (see Figure 1) which consists of an encoder, a structure decoder, and a cell decoder. The encoder captures the visual features of input table images. The structure decoder reconstructs table structure and helps the cell decoder to recognize cell content. Our EDD model is trained on PubTabNet and demonstrates superior performance compared to existing table recognition methods. The error analysis shows potential enhancements to the current EDD model for improved performance.

c) Evaluation: By modeling tables as a tree structure, we propose a new tree-edit-distance-based evaluate metric for image-based table recognition. This metric is superior to the metric [3] commonly used in literature and competitions, which completely ignores empty cells, information about the global table structure, and fine-grained performance on cell content recognition.

II. Related work

a) Data: Analyzing tabular data in unstructured documents focuses mainly on three problems: i) *table detection*: localizing the bounding boxes of tables in documents, ii) *table structure recognition*: parsing only the structural (row and column layout) information of tables, and iii) *table recognition*: parsing both the structural information and content of table cells. Table I compares the datasets that have been developed to address one or more of these three problems.

[1]https://github.com/ibm-aur-nlp/PubTabNet

[2]https://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/

The PubTabNet dataset and the EDD model we develop in this paper aim at the image-based table recognition problem. Comparing to other existing datasets for table recognition (*e.g*. SciTSR[3] and Table2Latex [4]), PubTabNet has three key advantages:

1) The tables are typeset by the publishers of the journals in PMCOA, which offers heterogeneous table styles.
2) Cells are categorized into headers and body cells, which is important when retrieving information from tables.
3) The format of targeted output is HTML, which can be directly integrated into web applications. In addition, tables in HTML format are represented as a tree structure. This enables the new tree-edit-distance-based evaluation metric that we propose in Section V

Dataset	TD	TSR	TR	# tables
Marmot [5]	✓	✗	✗	958
PubLayNet [6]	✓	✗	✗	113k
DeepFigures [7]	✓	✗	✗	1.4m
ICDAR2013 [2]	✓	✓	✓	156
ICDAR2019 [8]	✓	✓	✗	3.6k
UNLV [9]	✓	✓	✗	558
TableBank[4]	✓	✓	✗	417k (TD) 145k (TSR)
SciTSR[3]	✗	✓	✓	15k
Table2Latex [4]	✗	✓	✓	450k
PubTabNet	✗	✓	✓	568k

TABLE I: Datasets for Table Detection (TD), Table Structure Recognition (TSR) and Table Recognition (TR).

b) Model: Traditional table detection and recognition methods rely on pre-defined rules [10]–[14] and statistical machine learning [15]–[19]. Recently, deep learning exhibit great performance in image-based table detection and structure recognition. Hao *et al*. used a set of primitive rules to propose candidate table regions and a convolutional neural network to determine whether the regions contain a table [20]. Fully-convolutional neural networks, followed by a conditional random field, have also been used for table detection [21], [22]. In addition, deep neural networks for object detection, such as Faster-RCNN [23], Mask-RCNN [24], and YOLO [25] have been exploited for table detection and row/column segmentation [6], [26]–[28]. Furthermore, Qasim *et al*. used graph neural networks for table detection by encoding document images as graphs [29].

There are several tools (see Table II) that can convert tables in text-based PDF format into structured representations. However, there is limited work on image-based table recognition. Attention-based encoder-decoder was first proposed by Xu *et al*. for image captioning [30]. Deng *et al*. extended it by adding a recurrent layer in the encoder for capturing long horizontal spatial dependencies to convert images of mathematical formulas into LaTeX representation [31]. The same model was trained on the Table2Latex [4] dataset to convert table images into LaTeX representation. As show in [4] and in our experimental results (see Table II), the efficacy of this model on image-based table recognition is mediocre.

In this paper, we considerably improve the performance of the attention-based encoder-decoder method on image-based table recognition with a novel EDD architecture. Our model differs from other existing EDD architectures [32], [33], where the dual decoders are independent from each other. In our model, the cell decoder is triggered only when the structure decoder generates a new cell. In the meanwhile, the hidden state of the structure decoder is sent to the cell decoder to help it place its attention on the corresponding cell in the table image.

c) Evaluation: The evaluation metric proposed by Hurst is commonly used for table recognition, where the ground truth and recognition result of a table are flattened into a list of pairwise adjacency relations between non-empty cells [3]. Then precision and recall can be computed by comparing the recognized relations with the ground truth relations. Recognized relations are considered as true positive if an identical[5] relation can be found in the ground truth. Unmatched ground truth and recognized relations are treated false negative and false positive, respectively. This simple metric captures local similarity between the recognition result and ground truth, where empty cells and information about the global structure are completely ignored. This metric also cannot quantify the fine-grained performance of cell content recognition from table images, because any level of optical character recognition (OCR) error leads to the same outcome (unmatched relations). In order to address these issues, we propose a new evaluation metric: **T**ree-**E**dit-**D**istance-based **S**imilarity (TEDS). TEDS takes into account the global tree structure, and can capture fine-grained performance on cell content recognition.

III. Automatic generation of PubTabNet

PMCOA contains over one million scientific articles in both unstructured (PDF) and structured (XML) formats. A large table recognition dataset can be automatically generated if the corresponding location of the table nodes in the XML can be found in the PDF. Zhong *et al*. has proposed an algorithm to match the the XML and PDF representations of the articles in PMCOA, which automatically generated the PubLayNet dataset for document layout analysis [6]. We use their algorithm to extract the table regions from the PDF for the tables nodes in the XML. The table regions are converted to images with a 72 pixels per inch (PPI) resolution. We use this low PPI setting to relax the requirement of our model for high-resolution input images. For each table image, the corresponding table node (in HTML format) is extracted from the XML as the ground truth annotation.

It is observed that the algorithm generates erroneous bounding boxes for some tables, hence we use a heuristic to automatically verify the bounding boxes. For each annotation, the text within the bounding box is extracted from the PDF and compared with that in the annotation. The bounding box

[3]https://github.com/Academic-Hammer/SciTSR

[4]https://github.com/doc-analysis/TableBank

[5]Both cells are identical and the direction matches

is considered to be correct if the cosine similarity of the term frequency-inverse document frequency (Tf-idf) features of the two texts is greater than 90% and the length of the two texts differs less than 10%. In addition, to improve the learnability of the data, we remove rare tables which contains any cell that spans over 10 rows or 10 columns, or any character that occurs less than 50 times in all the tables. Tables of which the annotation contains `math` and `inline-formula` nodes are also removed, as we found they do not have a consistent XML representation.

After filtering the table samples, we curate the HTML code of the tables to remove unnecessary variations. First, we remove the nodes and attributes that are not reconstructable from the table image, such as hyperlinks and definition of acronyms. Second, table header cells are defined as `th` nodes in some tables, but as `td` nodes in others. We unify the definition of header cells as `td` nodes, which preserves the header identify of the cells as they are still descendants of the `thead` node. Third, all the attributes except 'rowspan' and 'colspan' in `td` nodes are stripped, since they control the appearance of the tables in web browsers, which do not match with the table image. These curations lead to consistent and clean HTML code and make the data more learnable.

Finally, the samples are randomly split into 60%/20%/20% training/development/test partitions. The training set contains 548,592 samples. As only a small proportion of tables contain spanning (multi-column or multi-row) cells, the evaluation on the raw development and test sets would be strongly biased towards tables without spanning cells. To better evaluate how a model performs on complex table structures, we create more balanced development and test sets by randomly drawing 5,000 tables with spanning cells and 5,000 tables without spanning cells from the corresponding raw set.

IV. Encoder-Dual-Decoder (EDD) Model

Figure 1 shows the architecture of the EDD model, which consists of an encoder, an attention-based structure decoder, and an attention-based cell decoder. The use of two decoders is inspired by two intuitive considerations: i) table structure recognition and cell content recognition are two distinctively

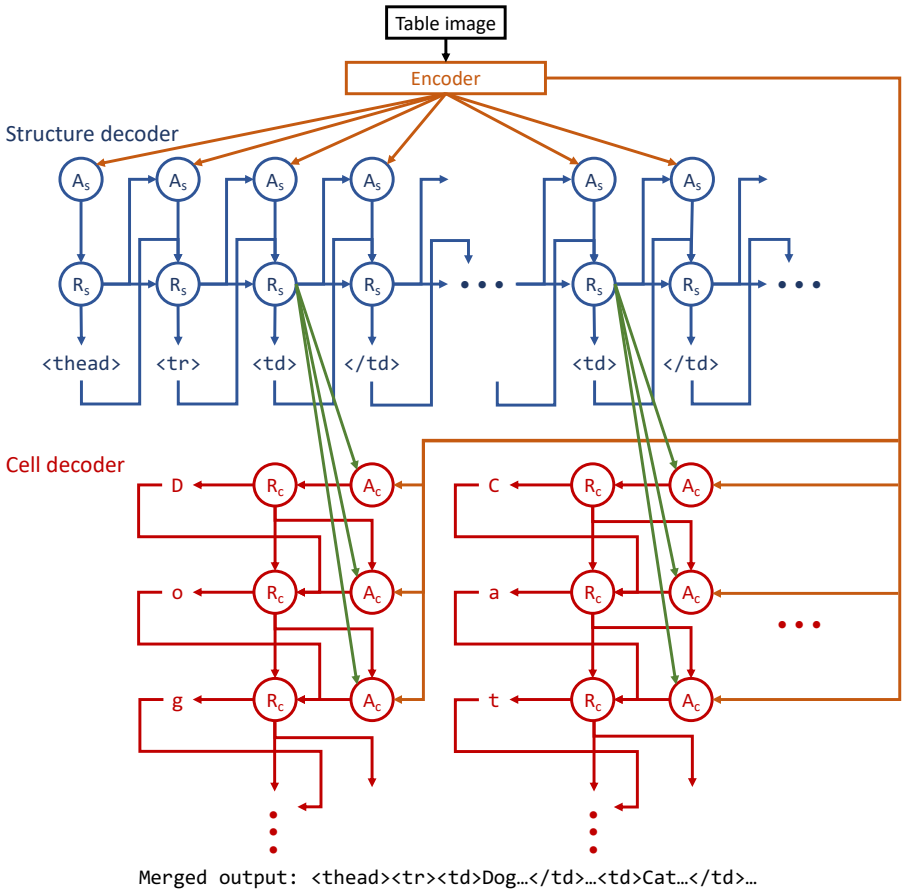


Fig. 1: EDD architecture. The encoder is a convolutional neural network which captures the visual features of the input table image. A_s and A_c are attention network for the structure decoder and cell decoder, respectively. R_s and R_c are recurrent units for the structure decoder and cell decoder, respectively. The structure decoder reconstructs table structure and helps the cell decoder to generate cell content. The output of the structure decoder and the cell decoder is merged to obtain the HTML representation of the input table image.

different tasks. It is not effective to solve both tasks at the same time using a single attention-based decoder. ii) information in the structure recognition task can be helpful for locating the cells that need to be recognized. The encoder is a convolutional neural network (CNN) that captures the visual features of input table images. The structure decoder and cell decoder are recurrent neural networks (RNN) with the attention mechanism proposed in [30]. The structure decoder only generates the HTML tags that define the structure of the table. When the structure decoder recognizes a new cell, the cell decoder is triggered and uses the hidden state of the structure decoder to compute the attention for recognizing the content of the new cell. This ensures a one-to-one match between the cells generated by the structure decoder and the sequences generated by the cell decoder. The outputs of the two decoders can be easily merged to get the final HTML representation of the table.

As the structure and the content of an input table image are recognized separately by two decoders, during training, the ground truth HTML representation of the table is tokenized into structural tokens, and cell tokens as shown in Figure 2. Structural tokens include the HTML tags that control the structure of the table. For spanning cells, the opening tag is broken down into multiple tokens as '<td', 'rowspan' or 'colspan' attributes, and '>'. The content of cells is tokenized at the character level, where HTML tags are treated as single tokens.

Two loss functions can be computed from the EDD network: i) cross-entropy loss of generating the structural tokens (l_s); and ii) cross-entropy loss of generating the cell tokens (l_c). The overall loss (l) of the EDD network is calculated as,

$$l = \lambda l_s + (1 - \lambda) l_c, \tag{1}$$

where $\lambda \in [0, 1]$ is a hyper-parameter.

V. Tree-edit-distance-based similarity (TEDS)

Tables are presented as a tree structure in the HTML format. The root has two children `thead` and `tbody`, which group table headers and table body cells, respectively. The children of `thead` and `tbody` nodes are table rows (`tr`). The leaves of the tree are table cells (`td`). Each cell node has three attributes, *i.e.* 'colspan', 'rowspan', and 'content'. We measure the similarity between two tables using the tree-edit distance proposed by Pawlik and Augsten [34]. The cost of insertion and deletion operations is 1. When the edit is substituting a node n_o with n_s, the cost is 1 if either n_o or n_s is not `td`. When both n_o and n_s are `td`, the substitution cost is 1 if the column span or the row span of n_o and n_s is different. Otherwise, the substitution cost is the normalized Levenshtein similarity [35] ($\in [0, 1]$) between the content of n_o and n_s. Finally, TEDS between two trees is computed as

$$TEDS(T_a, T_b) = 1 - \frac{EditDist(T_a, T_b)}{max(|T_a|, |T_b|)}, \tag{2}$$

where $EditDist$ denotes tree-edit distance, and $|T|$ is the number of nodes in T. The table recognition performance of a method on a set of test samples is defined as the mean of the TEDS score between the recognition result and ground truth of each sample.

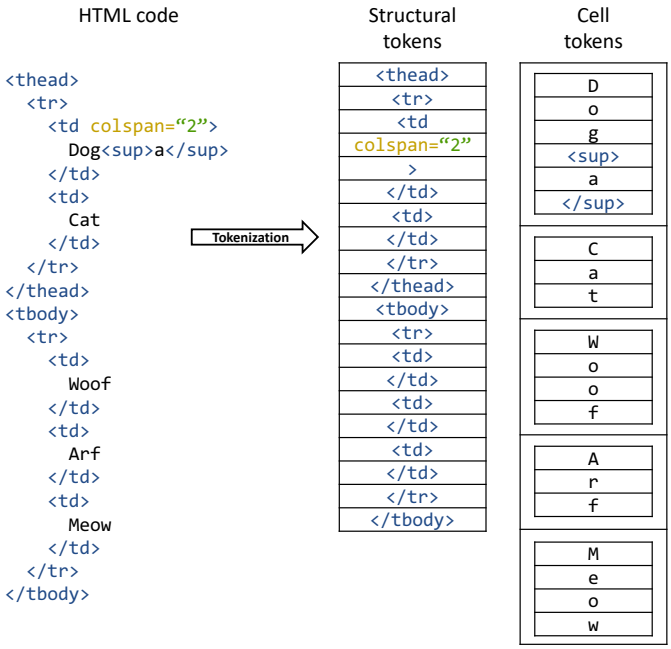


Fig. 2: Example of tokenizing a HTML table. Structural tokens define the structure of the table. HTML tags in cell content are treated as single tokens. The rest cell content is tokenized at the character level.

VI. Experiments

The test performance of the proposed EDD model is compared with five off-the-shelf tools (Tabula[6], Traprange[7], Camelot[8], PDFPlumber[9], and Adobe Acrobat® Pro[10]) and the WYGIWYS model[11] [31]. We crop the test tables from the original PDF for Tabula, Traprange, Camelot, and PDFPlumber, as they only support text-based PDF as input. Adobe Acrobat® Pro is tested with both PDF tables and high-resolution table images (300 PPI). The outputs of the off-the-shelf tools are parsed into the same tree structure as the HTML tables to compute the TEDS score.

A. Implementation details

To avoid exceeding GPU RAM, the EDD model is trained on a subset (399k samples) of PubTabNet training set, which

[6] v1.0.4 (https://github.com/tabulapdf/tabula-java)
[7] v1.0 (https://github.com/thoqbk/traprange)
[8] v0.7.3 (https://github.com/camelot-dev/camelot)
[9] v0.6.0-alpha (https://github.com/jsvine/pdfplumber)
[10] v2019.012.20040
[11] WYGIWYS is trained on the same samples as EDD by truncated back-propagation through time (200 steps), with a batch size of 10 and a learning rate 0.001 for 12 epochs and 0.0001 for another 3 epochs. The CNN in the encoder of WYGIWYS is replaced with the EDD-S1 setting, to rule out the possibility that the performance gain of EDD is due to different CNN.

satisfies

$$\begin{aligned} \text{width and height} &\leq 512 \text{ pixels} \\ \text{structural tokens} &\leq 300 \text{ tokens} \\ \text{longest cell} &\leq 100 \text{ tokens.} \end{aligned} \tag{3}$$

Note that samples in the validation and test sets are not constrained by these criteria. The vocabulary size of the structural tokens and the cell tokens of the training data is 32 and 281, respectively. Training images are rescaled to 448×448 pixels to facilitate batching and each channel is normalized by z-score.

We use the ResNet-18 [36] network as the encoder. The default ResNet-18 model downsamples the image resolution by 32. We modify the last CNN layer of ResNet-18 to study if a higher-resolution feature map improves table recognition performance. A total of five different settings are tested in this paper:

- EDD-S2: the default ResNet-18
- EDD-S1: stride of the last CNN layer set to 1
- EDD-S2S2: two independent last CNN layers for structure (stride=2) and cell (stride=2) decoder
- EDD-S2S1: two independent last CNN layers for structure (stride=2) and cell (stride=1) decoder
- EDD-S1S1: two independent last CNN layers for structure (stride=1) and cell (stride=1) decoder

We evaluate the performances of these five settings on the validation set and find that higher-resolution feature map and independent CNN layers improve performance. As a result, the EDD-S1S1 setting provides the best validation performance, and is therefore chosen to compare with baselines on the test set.

The structure decoder and the cell decoder are single-layer long short-term memory (LSTM) networks, of which the hidden state size is 256 and 512, respectively. Both of the decoders weight the feature map from the encoder with soft-attention, which has a hidden layer of size 256. The embedding dimension of structural tokens and cell tokens is 16 and 80, respectively. At inference time, the output of both of the decoders are sampled with beam search (beam=3).

The EDD model is trained with the Adam [37] optimizer with two stages. First, we pre-train the encoder and the structure decoder to generate the structural tokens only ($\lambda = 1$), where the batch size is 10, and the learning rate is 0.001 in the first 10 epochs and reduced by 10 for another 3 epochs. Then we train the whole EDD network to generate both structural and cell tokens ($\lambda = 0.5$), with a batch size 8 and a learning rate 0.001 for 10 epochs and 0.0001 for another 2 epochs. Total training time is about 16 days on two V100 GPUs.

B. Quantitative analysis

Table II compares the test performance of the proposed EDD model and the baselines, where the average TEDS of simple[12] and complex[13] test tables is also shown. By solely relying on table images, EDD substantially outperforms all the baselines on recognizing simple and complex tables, even the ones that directly use text extracted from PDF to fill table cells. Adobe Acrobat® Pro, when taking PDF as input, is the best off-the-shelf tool in this comparison. Nevertheless, its performance decreases dramatically when fed with images. When trained on the PubTabNet dataset, WYGIWYS also considerably outperform the off-the-shelf tools, but is outperformed by EDD by 9.7% absolute TEDS score. The advantage of EDD to WYGIWYS is more profound on complex tables (9.9% absolute TEDS) than simple tables (9.5% absolute TEDS). This proves the great advantage of jointly training two separate decoders to solve structure recognition and cell content recognition tasks.

Input	Method	Average TEDS (%)		
		Simple[12]	Complex[13]	All
PDF	Tabula	0.677	0.489	0.583
	Traprange	0.608	0.499	0.554
	Camelot	0.691	0.548	0.620
	PDFPlumber	0.383	0.292	0.338
	Acrobat® Pro	0.689	0.618	0.653
Image	Acrobat® Pro	0.538	0.535	0.537
	WYGIWYS	0.817	0.755	0.786
	EDD	0.912	0.854	0.883

TABLE II: Test performance of EDD and 7 baseline approaches. Our EDD model, by solely relying on table images, substantially outperforms all the baselines.

C. Qualitative analysis

To illustrate the differences in the behavior of the compared methods, Figure 3 shows the rendering of the predicted HTML given an example input table. The table has 7 columns, 3 header rows, and 4 body rows. The table header has a complex structure, which consists of 4 multi-row (span=3) cells, 2 multi-column (span=3) cells, and three normal cells. Our EDD model is able to generate an extremely close match to the ground truth, making no error in structure recognition and a single OCR error ('PF' recognized as 'PC'). The second header row is missing in the results of WYGIWYS, which also makes a few errors in the cell content. On the other hand, the off-the-shelf tools make substantially more errors in recognizing the complex structure of the table headers. This demonstrates the limited capability of these tools on recognizing complex tables.

Figures 4 (a) - (c) illustrate the attention of the structure decoder when processing an example input table. When a new row is recognized ('<tr>' and '</tr>'), the structure decoder focuses its attention around the cells in the row. When the opening tag ('<td>') of a new cell is generated, the structure decoder pays more attention around the cell. For the closing tag '</td>' tag, the attention of the structure decoder spreads across the image. Since '</td>' always follows the '<td>' or '>' token, the structure decoder relies on the language model rather than the encoded feature map to predict it. Figure 4 (d) shows the aggregated attention of the cell decoder when generating the content of each cell. Compared to the structure

[12]Tables without multi-column or multi-row cells.
[13]Tables with multi-column or multi-row cells.

Time after IVF (h)	No. of oocytes (replicates)	No. of MII oocytes (%)*	No. of fertilization (%)**	Embryo development (% of fertilized oocytes)		
				OA (%)	PF (%)	CC (%)
12	103 (9)	63 (61.2)	28.6[a]	5 (27.8)	13 (72.2)	0 (0)
18	97 (7)	65 (67.0)	50.8[b]	3 (9.1)	30 (90.9)	0 (0)
24	91 (7)	59 (64.9)	49.2[b]	4 (13.8)	25 (86.2)	0 (0)
30	87 (8)	56 (64.4)	48.2[b]	4 (14.9)	9 (33.3)	14 (51.8)

(a) Input table

Time after IVF (h)	**No. of oocytes (replicates)**	**No. of MII oocytes (%)***	**No. of fertilization (%)****	**Embryo development (% of fertilized oocytes)**		
				OA (%)	**PF (%)**	**CC (%)**
12	103 (9)	63 (61.2)	28.6[a]	5 (27.8)	13 (72.2)	0 (0)
18	97 (7)	65 (67.0)	50.8[b]	3 (9.1)	30 (90.9)	0 (0)
24	91 (7)	59 (64.9)	49.2[b]	4 (13.8)	25 (86.2)	0 (0)
30	87 (8)	56 (64.4)	48.2[b]	4 (14.9)	9 (33.3)	14 (51.8)

(b) Ground truth

Time after IVF (h)	**No. of oocytes (replicates)**	**No. of MII oocytes (%)***	**No. of fertilization (%)****	**Embryo development (% of fertilized oocytes)**		
				OA (%)	**PC (%)**	**CC (%)**
12	103 (9)	63 (61.2)	28.6[a]	5 (27.8)	13 (72.2)	0 (0)
18	97 (7)	65 (67.0)	50.8[b]	3 (9.1)	30 (90.9)	0 (0)
24	91 (7)	59 (64.9)	49.2[b]	4 (13.8)	25 (86.2)	0 (0)
30	87 (8)	56 (64.4)	48.2[b]	4 (14.9)	9 (33.3)	14 (51.8)

(c) EDD (TEDS = 99.8%)

Time after IV (%)	**No. of covering (regionales)**	**No. of MI total (*n* (%))**	**No. of fertilization (%)****	**Embryo development**		
				OA (%)	**PF (%)**	**CC (%)**
12	103 (9)	63 (61.2)	28.6[a]	5 (27.8)	13 (72.2)	0 (0)
18	97 (7)	65 (67.0)	50.8[b]	3 (9.1)	30 (90.9)	0 (0)
24	91 (7)	59 (64.9)	49.2[b]	4 (13.8)	25 (86.2)	0 (0)
30	87 (8)	56 (64.4)	48.2[b]	4 (14.9)	9 (33.3)	14 (51.8)

(d) WYGIWYS (TEDS = 89.8%)

Time after IVF (h) No. of oocytes No. of MIIâ			No. of Embryo developmentfertilization (% of fertilized oocytes)			
	(replicates)	oocytes (%)	(%)âã	OA (%)	PF (%)	CC (%)
12	103 (9)	63 (61.2)	28.6a	5 (27.8)	13 (72.2)	0 (0)
18	97 (7)	65 (67.0)	50.8b	3 (9.1)	30 (90.9)	0 (0)
24	91 (7)	59 (64.9)	49.2b	4 (13.8)	25 (86.2)	0 (0)
30	87 (8)	56 (64.4)	48.2b	4 (14.9)	9 (33.3)	14 (51.8)

(e) Adobe Acrobat® Pro on PDF (TEDS = 74.8%)

(replicates)	oocytes (o/o)* (o/o)**				
			OA (o/o)	PF(%)	CC(%)
12 103 (9)	63 (61.2)	28.6a	5 (27.8)	13 (72.2)	0 (O)
18 97 (7)	65 (67.0)	SO.Sb	3 (9.1)	30 (90.9)	0 (O)
24 91 (7)	59 (64.9)	49.2b	4(13.8)	25 (86.2)	0 (O)
30 87 (8)	56 (64.4)	48.2b	4 (14.9)	9 (33.3)	14 (51.8)

(f) Adobe Acrobat® Pro on Image (TEDS = 64.2%)

			No. of		**Embryo development**	
Time after IVF (h)	No. of oocytes	No. of MII	fertilization			
	(replicates)	oocytes (%)*			(% of fertilized oocytes)	
			(%)**			
				OA (%)	PF (%)	CC (%)
12	103 (9)	63 (61.2)	28.6a	5 (27.8)	13 (72.2)	0 (0)
18	97 (7)	65 (67.0)	50.8b	3 (9.1)	30 (90.9)	0 (0)
24	91 (7)	59 (64.9)	49.2b	4 (13.8)	25 (86.2)	0 (0)
30	87 (8)	56 (64.4)	48.2b	4 (14.9)	9 (33.3)	14 (51.8)

(g) Tabula (TEDS = 42.6%)

Time	after	IVF	(h)	N(roep.loifcaotoecs)ytes	Noooc.yotfesM(%II)ã	No.of f(e%rt)iãliãzation		Embryodevelopment (%offertilizedoocytes)	
							OA(%)	PF(%)	CC(%)
				103(9)	63(61.2)	28.6a	5(27.8)	13(72.2)	0(0)
				97(7)	65(67.0)	50.8b	3(9.1)	30(90.9)	0(0)
				91(7)	59(64.9)	49.2b	4(13.8)	25(86.2)	0(0)
				87(8)	56(64.4)	48.2b	4(14.9)	9(33.3)	14(51.8)

(h) Traprange (TEDS = 40.2%)

			No. of		Embryo development	
	No. of oocytes	No. of MII	fertilization			
Time after IVF (h)					(% of fertilized oocytes)	
	(replicates)	oocytes (%)*				
			(%)**			
				OA (%)	PF (%)	CC (%)
12	103 (9)	63 (61.2)	28.6a	5 (27.8)	13 (72.2)	0 (0)
18	97 (7)	65 (67.0)	50.8b	3 (9.1)	30 (90.9)	0 (0)
24	91 (7)	59 (64.9)	49.2b	4 (13.8)	25 (86.2)	0 (0)
30	87 (8)	56 (64.4)	48.2b	4 (14.9)	9 (33.3)	14 (51.8)

(i) Camelot (TEDS = 31.7%)

	No.ofoocytes	**No.ofMII**				
	(replicates)	oocytes(%)*				
				OA(%)	PF(%)	CC(%)
12	103(9)	63(61.2)	28.6a	5(27.8)	13(72.2)	0(0)
18	97(7)	65(67.0)	50.8b	3(9.1)	30(90.9)	0(0)
24	91(7)	59(64.9)	49.2b	4(13.8)	25(86.2)	0(0)
30	87(8)	56(64.4)	48.2b	4(14.9)	9(33.3)	14(51.8)

(j) PDFPlumber (TEDS = 28.8%)

Fig. 3: Table recognition results of EDD and 7 baseline approaches on an example input table which has a complex header structure (4 multi-row (span=3) cells, 2 multi-column (span=3) cells, and three normal cells). Our EDD model perfectly recognizes the complex structure and cell content of the table, whereas the baselines struggle with the complex table header.

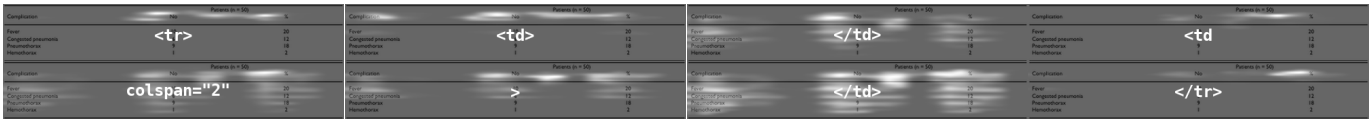
decoder, the cell decoder has more focused attention, which falls on the cell content that is being generated.

D. Error analysis

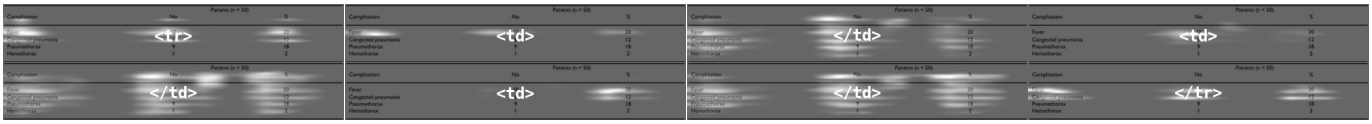
We categorize the test set of PubTabNet into 15 equal-interval groups along four key properties of table size: width, height, number of structural tokens, and number of tokens in the longest cell. Figure 5 illustrates the number of tables in each group and the performance of the EDD model and the WYGIWYS model on each group. The EDD model outperforms the WYGIWYS model on all groups. The performance of both models decreases as table size increases. We train the models with tables that satisfy Equation 3, where the thresholds are indicated with vertical dashed lines in Figure 5. Except for width, we do not observe a steep decrease in performance near the thresholds. We think the lower performance on larger tables is mainly due to rescaling images for batching, where larger tables are more strongly downsampled. The EDD model may better handle large tables by grouping table images into similar sizes as in [31] and using different rescaling sizes for each group.

VII. Conclusion

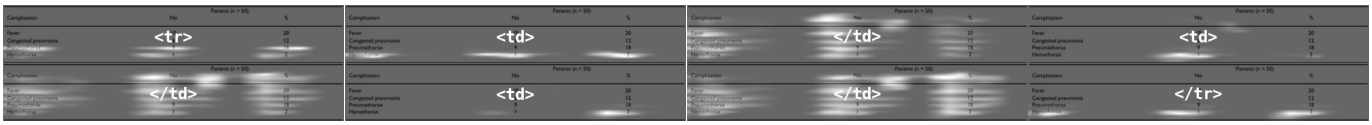
This paper makes a comprehensive study of the image-based table recognition problem. A large-scale dataset PubLayNet is developed to train and evaluate deep learning models.



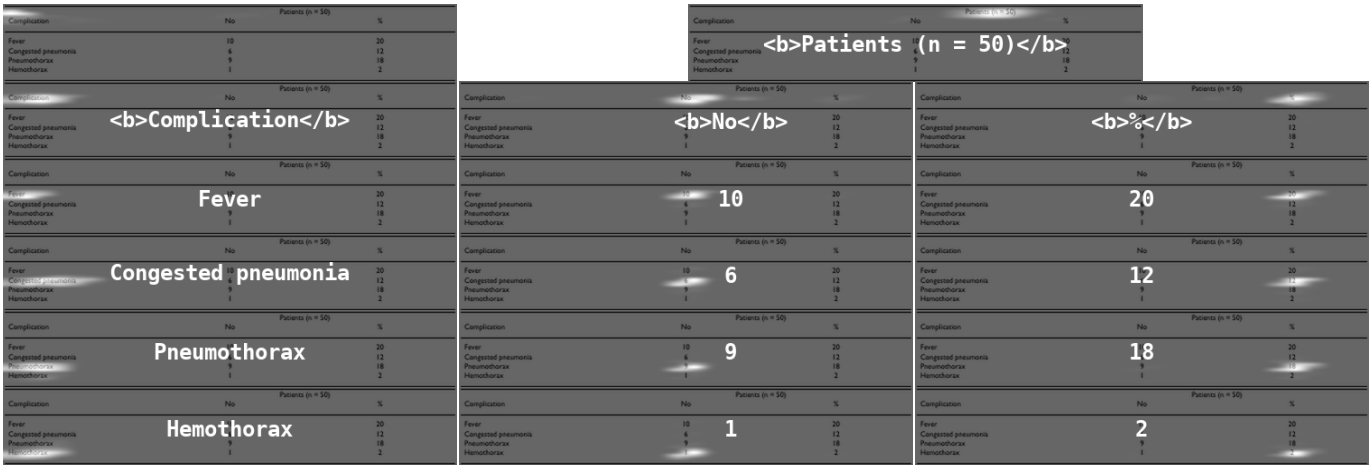
(a) Attention of structure decoder on the first header row



(b) Attention of structure decoder on the first body row



(c) Attention of structure decoder on the last body row



(d) Aggregated attention of cell decoder on each cell

Fig. 4: Attention distribution of the structure decoder (a - c) and the cell decoder (d) on an example input table. The texts at the center of the images are predictions of the EDD model. The structure decoder focuses its attention around table cells when recognizing new rows and cells, whereas the cell decoder places more focused attention on cell content.

By separating table structure recognition and cell content recognition tasks, we propose an attention-based EDD model. The structure decoder not only recognizes the structure of input tables, but also helps the cell decoder to place its attention on the right cell content. We also propose a new evaluation metric TEDS, which captures both the performance of table structure recognition and cell content recognition. The proposed EDD model, when trained on PubLayNet, is effective on recognizing complex table structures and extracting cell content from image. PubTabNet has been made available and we believe that PubTabNet will accelerate future development in table recognition and provide support for pre-training table recognition models.

Our future works will focus on the following two directions. First, current PubTabNet dataset does not provide coordinates of table cells, which we plan to supplement in the next version. This will enable adding an additional branch to the EDD network to also predict cell location. We think this additional task will assist cell content recognition. In addition, when tables are available in text-based PDF format, the cell location can be used to extract cell content directly from PDF without using OCR, which might improve the overall recognition quality. Second, the EDD model takes table images as input, which implicitly assumes that the accurate location of tables in documents is given by users. We will investigate how the EDD model can be integrated with table detection neural networks to achieve end-to-end table detection and recognition.

REFERENCES

[1] A. Jimeno Yepes and K. Verspoor, "Literature mining of genetic variants for curation: quantifying the importance of supplementary material," *Database*, vol. 2014, 2014.

[2] M. Göbel, T. Hassan, E. Oro, and G. Orsi, "ICDAR 2013 table competition," in *2013 12th International Conference on Document Analysis and Recognition*. IEEE, 2013, pp. 1449–1453.

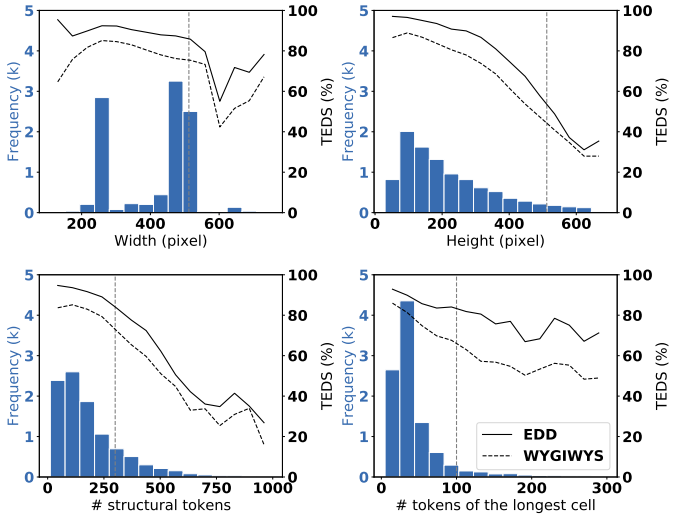


Fig. 5: Dependence of the performance of the EDD model and the WYGIWYS model on table size in terms of width, height, number of structural tokens, and number of tokens in the longest cell. The bar plots (left axis) are the histogram of PubTabNet test set w.r.t. the above four properties. The line plots (right axis) are the mean TEDS score of the test tables in each bar. The vertical dashed lines are the thresholds in Equation 3.

[3] M. Hurst, "A constraint-based approach to table structure derivation," 2003.

[4] Y. Deng, D. Rosenberg, and G. Mann, "Challenges in end-to-end neural scientific table recognition," in *2019 15th International Conference on Document Analysis and Recognition*. IEEE, 2019, p. accepted.

[5] J. Fang, X. Tao, Z. Tang, R. Qiu, and Y. Liu, "Dataset, ground-truth and performance metrics for table detection evaluation," in *2012 10th IAPR International Workshop on Document Analysis Systems*. IEEE, 2012, pp. 445–449.

[6] X. Zhong, J. Tang, and A. J. Yepes, "Publaynet: largest dataset ever for document layout analysis," in *2019 15th International Conference on Document Analysis and Recognition*. IEEE, 2019, p. accepted.

[7] N. Siegel, N. Lourie, R. Power, and W. Ammar, "Extracting scientific figures with distantly supervised neural networks," in *Proceedings of the 18th ACM/IEEE on joint conference on digital libraries*. ACM, 2018, pp. 223–232.

[8] L. Gao, Y. Huang, Y. Li, Q. Yan, Y. Fang, H. Dejean, F. Kleber, and E.-M. Lang, "ICDAR 2019 competition on table detection and recognition," in *2019 15th International Conference on Document Analysis and Recognition*. IEEE, 2019, p. accepted.

[9] A. Shahab, F. Shafait, T. Kieninger, and A. Dengel, "An open approach towards the benchmarking of table structure recognition systems," in *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*. ACM, 2010, pp. 113–120.

[10] Y. Hirayama, "A method for table structure analysis using dp matching," in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, vol. 2. IEEE, 1995, pp. 583–586.

[11] S. Tupaj, Z. Shi, C. H. Chang, and H. Alam, "Extracting tabular information from text files," *EECS Department, Tufts University, Medford, USA*, 1996.

[12] J. Hu, R. S. Kashi, D. P. Lopresti, and G. Wilfong, "Medium-independent table detection," in *Document Recognition and Retrieval VII*, vol. 3967. International Society for Optics and Photonics, 1999, pp. 291–302.

[13] B. Gatos, D. Danatsas, I. Pratikakis, and S. J. Perantonis, "Automatic table detection in document images," in *International Conference on Pattern Recognition and Image Analysis*. Springer, 2005, pp. 609–618.

[14] F. Shafait and R. Smith, "Table detection in heterogeneous documents," in *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*. ACM, 2010, pp. 65–72.

[15] T. Kieninger and A. Dengel, "The t-recs table recognition and analysis system," in *International Workshop on Document Analysis Systems*. Springer, 1998, pp. 255–270.

[16] F. Cesarini, S. Marinai, L. Sarti, and G. Soda, "Trainable table location in document images," in *Object recognition supported by user interaction for service robots*, vol. 3. IEEE, 2002, pp. 236–240.

[17] A. C. e Silva, "Learning rich hidden markov models in document analysis: Table location," in *2009 10th International Conference on Document Analysis and Recognition*. IEEE, 2009, pp. 843–847.

[18] T. Kasar, P. Barlas, S. Adam, C. Chatelain, and T. Paquet, "Learning to detect tables in scanned document images using line information," in *2013 12th International Conference on Document Analysis and Recognition*. IEEE, 2013, pp. 1185–1189.

[19] M. Fan and D. S. Kim, "Table region detection on large-scale pdf files without labeled data," *CoRR, abs/1506.08891*, 2015.

[20] L. Hao, L. Gao, X. Yi, and Z. Tang, "A table detection method for pdf documents based on convolutional neural networks," in *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*. IEEE, 2016, pp. 287–292.

[21] D. He, S. Cohen, B. Price, D. Kifer, and C. L. Giles, "Multi-scale multi-task fcn for semantic page segmentation and table detection," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1. IEEE, 2017, pp. 254–261.

[22] I. Kavasidis, C. Pino, S. Palazzo, F. Rundo, D. Giordano, P. Messina, and C. Spampinato, "A saliency-based convolutional neural network for table and chart detection in digitized documents," in *International Conference on Image Analysis and Processing*. Springer, 2019, pp. 292–302.

[23] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[24] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[25] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[26] S. Schreiber, S. Agne, I. Wolf, A. Dengel, and S. Ahmed, "Deepdesrt: Deep learning for detection and structure recognition of tables in document images," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1. IEEE, 2017, pp. 1162–1167.

[27] A. Gilani, S. R. Qasim, I. Malik, and F. Shafait, "Table detection using deep learning," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1. IEEE, 2017, pp. 771–776.

[28] P. W. Staar, M. Dolfi, C. Auer, and C. Bekas, "Corpus conversion service: A machine learning platform to ingest documents at scale," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018, pp. 774–782.

[29] S. R. Qasim, H. Mahmood, and F. Shafait, "Rethinking table recognition using graph neural networks."

[30] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*, 2015, pp. 2048–2057.

[31] Y. Deng, A. Kanervisto, J. Ling, and A. M. Rush, "Image-to-markup generation with coarse-to-fine attention," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 980–989.

[32] Y.-F. Zhou, R.-H. Jiang, X. Wu, J.-Y. He, S. Weng, and Q. Peng, "Branchgan: Unsupervised mutual image-to-image transfer with a single encoder and dual decoders," *IEEE Transactions on Multimedia*, 2019.

[33] R. Morais, V. Le, T. Tran, B. Saha, M. Mansour, and S. Venkatesh, "Learning regularity in skeleton trajectories for anomaly detection in videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 996–12 004.

[34] M. Pawlik and N. Augsten, "Tree edit distance: Robust and memory-efficient," *Information Systems*, vol. 56, pp. 157–173, 2016.

[35] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," in *Soviet physics doklady*, vol. 10, no. 8, 1966, pp. 707–710.

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2015.