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

Xu Zhong[1[0000−0002−0619−8949]], Elaheh ShafieiBavani[1[0000−0001−8546−1217]], and Antonio Jimeno Yepes[1[0000−0002−6581−094X]]

IBM Research Australia, 60 City Road, Southgate, VIC 3006, Australia
peter.zhong@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 and release 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, which more appropriately captures multi-hop cell misalignment and OCR errors than the pre-established metric. 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.

Keywords: table recognition, dual decoder, dataset, evaluation

1 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

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

the narrative part of the article compared to the information available in tables or files in tabular format [17].

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* [9].

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

- **Data** We provide a large-scale dataset PubTabNet, which consists of over 568k images and corresponding HTML representations of heterogeneous tables. PubTabNet is created by matching the PDF format and the XML format of the scientific articles contained in PMCOA[2].
- **Model** We develop a novel attention-based encoder-dual-decoder (EDD) architecture (see Fig. 1) which consists of an encoder, a structure decoder, and a cell decoder. The EDD model is the first end-to-end table recognition model that supports joint training on table structure recognition and cell content recognition tasks. This model design allows the cell decoder to use information from the structure decoder to better focus on the local visual features of the cell being generated. This mechanism back-propagates cell content recognition loss to the structure decoder, which regularizes it to better locate table cells. EDD demonstrates superior performance on PubTabNet, compared to existing table recognition methods.
- **Evaluation** By modeling tables as a tree structure, we propose a new tree-edit-distance-based evaluate metric for image-based table recognition. We demonstrate that our new metric is superior to the metric [16] commonly used in literature and competitions.

2 Related work

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 1 compares the datasets that have been developed to address one or more of these three problems. 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

[2]https://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/
[3]https://github.com/doc-analysis/TableBank

Dataset	TD	TSR	TR	# tables
Marmot [5]	✓	✗	✗	958
PubLayNet [39]	✓	✗	✗	113k
DeepFigures [33]	✓	✗	✗	1.4m
ICDAR2013 [9]	✓	✓	✓	156
ICDAR2019 [6]	✓	✓	✗	3.6k
UNLV [32]	✓	✓	✗	558
TableBank[3]	✓	✓	✗	417k (TD) 145k (TSR)
SciTSR[4]	✗	✓	✓	15k
Table2Latex [3]	✗	✓	✓	450k
Synthetic data in [26]	✗	✓	✓	Unbounded
PubTabNet	✗	✓	✓	568k

Table 1: Datasets for Table Detection (TD), Table Structure Recognition (TSR) and Table Recognition (TR).

table recognition (*e.g.* SciTSR[4], Table2Latex [3], and TIES [26]), PubTabNet has three key advantages:

1. The tables are typeset by the publishers of over 6,000 journals, which offers considerably more diversity in table styles than other table datasets.
2. Cells are categorized into headers and body cells, which is important when retrieving information from tables.

Model Traditional table detection and recognition methods rely on pre-defined rules [7, 14, 15, 24, 31, 37] and statistical machine learning [1, 4, 18, 20, 34]. 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 [10]. Fully-convolutional neural networks, followed by a conditional random field, have also been used for table detection [11,19,36]. In addition, deep neural networks for object detection, such as Faster-RCNN [28], Mask-RCNN [12], and YOLO [27] have been exploited for table detection and row/column segmentation [8, 30, 35, 39]. Furthermore, graph neural networks are used for table detection and recognition by encoding document images as graphs [26, 29].

There are several tools (see Table 2) 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 [38]. 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 [2]. The

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

same model was trained on the Table2Latex [3] dataset to convert table images into LaTeX representation. As show in [3] and in our experimental results (see Table 2), the efficacy of this model on image-based table recognition is mediocre.

This paper considerably improves 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 [23, 40], 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.

Evaluation The evaluation metric proposed in [16] is commonly used in table recognition literature and competitions. This metric first flattens the ground truth and recognition result of a table are into a list of pairwise adjacency relations between non-empty cells. Then precision, recall, and F1-score can be computed by comparing the lists. This metric is simple but has two obvious problems: 1) as it only checks immediate adjacency relations between non-empty cells, it cannot detect errors caused by empty cells and misalignment of cells beyond immediate neighbors; 2) as it checks relations by exact match[5], it does not have a mechanism to measure fine-grained cell content recognition performance. In order to address these two problems, we propose a new evaluation metric: **T**ree-**E**dit-**D**istance-based **S**imilarity (TEDS). TEDS solves problem 1) by examining recognition results at the global tree-structure level, allowing it to identify all types of structural errors; and problem 2) by computing the string-edit-distance when the tree-edit operation is node substitution.

3 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. In our previous work, we proposed an algorithm to match the the XML and PDF representations of the articles in PMCOA [39]. We use this 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 (HTML) 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 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

[5]Both cells are identical and the direction matches

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 partitioned into 60%/20%/20% training/development/test sets. 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.

4 Encoder-dual-decoder (EDD) model

Fig. 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 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 [38]. 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

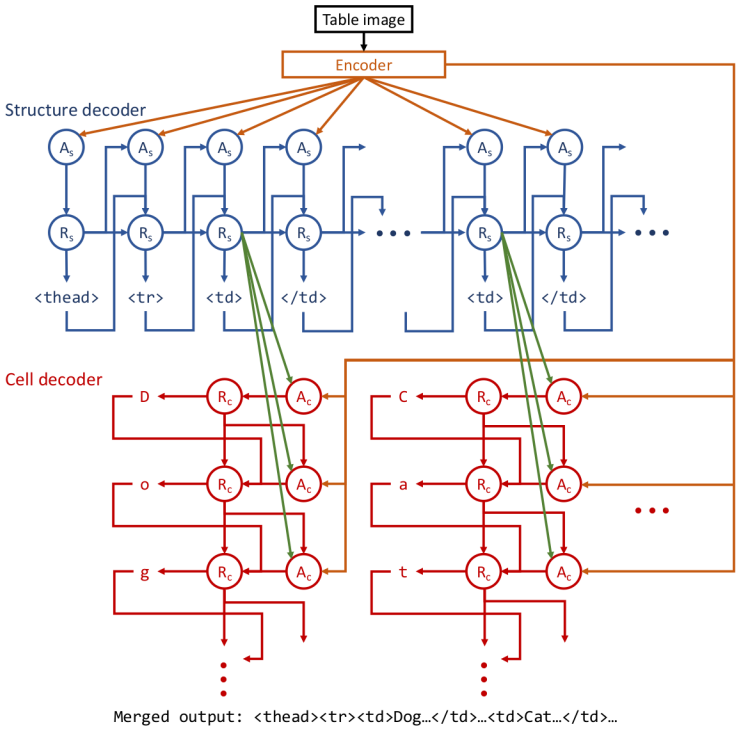


Fig. 1: EDD architecture. The encoder is a convolutional neural network which captures the visual features of the input table image. A_s and A_c are attention network for the structure decoder and cell decoder, respectively. R_s and R_c are recurrent units for the structure decoder and cell decoder, 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.

of the table is tokenized into structural tokens, and cell tokens as shown in Fig. 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.

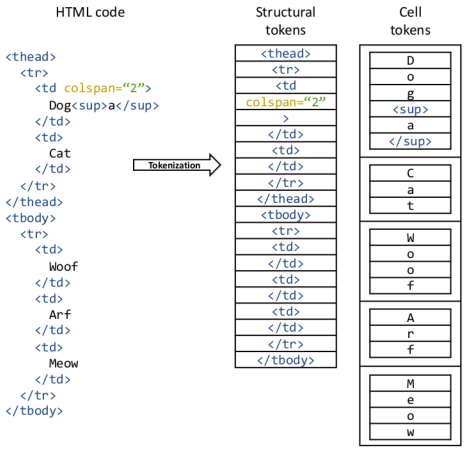


Fig. 2: Example of tokenizing a HTML table. Structural tokens define the structure of the table. HTML tags in cell content are treated as single tokens. The rest cell content is tokenized at the character level.

5 Tree-edit-distance-based similarity (TEDS)

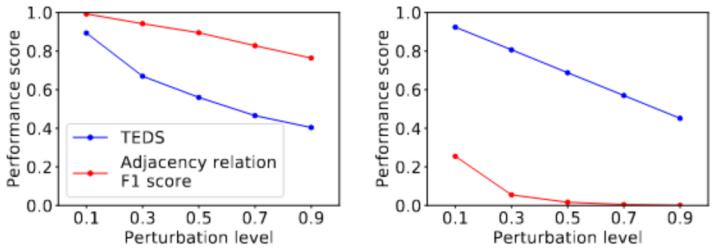
Tables are presented as a tree structure in the HTML format. The root has two children `thead` and `tbody`, which group table headers and table body cells, respectively. The children of `thead` and `tbody` nodes are table rows (`tr`). The leaves of the tree are table cells (`td`). Each cell node has 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 [25]. 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 [22] ($\in [0, 1]$) 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.

In order to justify that TEDS solves the two problems of the adjacency relation metric [16] described previously in Section 2, we add two types of perturbations to the validation set of PubTabNet and examine how TEDS and the adjacency relation metric respond to the perturbations.

1. To demonstrate the empty-cell and multi-hop misalignment issue, we shift some cells in the first row downwards[6], and pad the leftover space with empty cells. The shift distance of a cell is proportional to its column index. We tested 5 perturbation levels, i.e., 10%, 30%, 50%, 70%, or 90% of the cells in the first row are shifted. Fig. S1 in supplemental material shows a perturbed example, where 90% of the cells in the first row are shifted.
2. To demonstrate the fine-grained cell content recognition issue, we randomly modify some characters into a different one. We tested 5 perturbation levels, i.e., the chance that a character gets modified is set to be 10%, 30%, 50%, 70%, or 90%. Fig. S2 in supplemental material shows an example at the 10% perturbation level.



(a) Cell shift perturbation (b) Cell content perturbation

Fig. 3: Comparison of the response of TEDS and the adjacency relation metric to cell shift perturbation and cell content perturbation. The adjacency relation metric is under-reacting to cell shift perturbation and over-reacting to cell content perturbation. Whereas TEDS demonstrates superiority at appropriately capturing the errors.

Fig. 3 illustrates how TEDS and the adjacency relation F1-score respond to the two types of perturbations at different levels. The adjacency relation metric is under-reacting to the cell shift perturbation. At the 90% perturbation level, the table is substantially different from the original (see example in Fig. S1 in supplemental material). However, the adjacency relation F1-score is still nearly 80%. On the other hand, the perturbation causes a 60% drop on TEDS, demonstrating that TEDS is able to capture errors that the adjacency relation metric cannot.

When it comes to cell content perturbations, the adjacency relation metric is over-reacting. Even the 10% perturbation level (see example in Fig. S2 in supplemental material) leads to over 70% decrease in adjacency relation F1-score, which drops close to zero from the 50% perturbation level. In contrast,

[6]If the number of rows is greater than the number of columns, we shift the cells in the first column rightwards instead.

TEDS linearly decreases from 90% to 40% as the perturbation level increases from 10% to 90%, demonstrating the capability of capturing fine-grained cell content recognition errors.

6 Experiments

The test performance of the proposed EDD model is compared with five off-the-shelf tools (Tabula[7], Traprange[8], Camelot[9], PDFPlumber[10], and Adobe Acrobat® Pro[11]) and the WYGIWYS model[12] [2]. 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.

6.1 Implementation details

To avoid exceeding GPU RAM, the EDD model is trained on a subset (399k samples) of PubTabNet training set, which 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 [13] 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

[7] v1.0.4 (https://github.com/tabulapdf/tabula-java)
[8] v1.0 (https://github.com/thoqbk/traprange)
[9] v0.7.3 (https://github.com/camelot-dev/camelot)
[10] v0.6.0-alpha (https://github.com/jsvine/pdfplumber)
[11] v2019.012.20040
[12] WYGIWYS is trained on the same samples as EDD by truncated back-propagation through time (200 steps). WYGIWYS and EDD use the same CNN in the encoder to rule out the possibility that the performance gain of EDD is due to difference in CNN.

- 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 (see Table S3 in supplemental material) and find that a 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 [21] 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.

6.2 Quantitative analysis

Table 2 compares the test performance of the proposed EDD model and the baselines, where the average TEDS of simple[13] and complex[14] 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. Camelot is the best off-the-shelf tool in this comparison. Furthermore, the performance of Adobe Acrobat® Pro on image input is dramatically lower than that on PDF input, demonstrating the difficulty of recognizing tables solely on table 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.

6.3 Qualitative analysis

To illustrate the differences in the behavior of the compared methods, Fig. 4 shows the rendering of the predicted HTML given an example input table. The

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

Input	Method	Average TEDS (%)		
		Simple[13]	Complex[14]	All
PDF	Tabula	78.0	57.8	67.9
	Traprange	60.8	49.9	55.4
	Camelot	80.0	66.0	73.0
	PDFPlumber	44.9	35.9	40.4
	Acrobat® Pro	68.9	61.8	65.3
Image	Acrobat® Pro	53.8	53.5	53.7
	WYGIWYS	81.7	75.5	78.6
	EDD	**91.2**	**85.4**	**88.3**

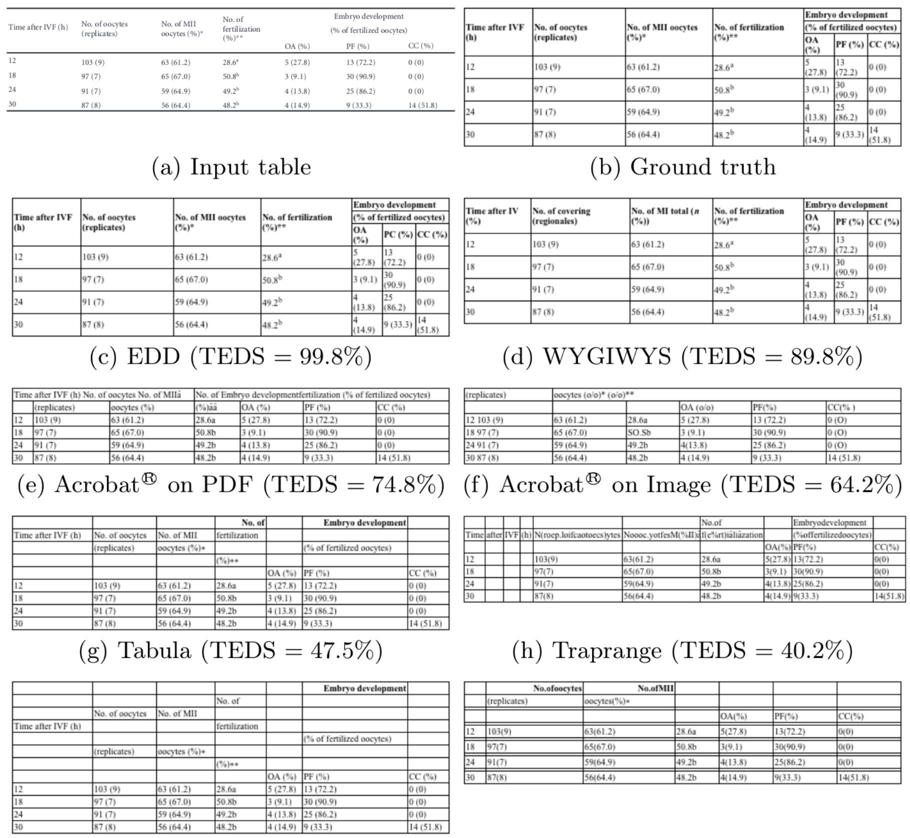
Table 2: Test performance of EDD and 7 baseline approaches. Our EDD model, by solely relying on table images, substantially outperforms all the baselines.

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 optical character recognition (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.

Figs. S4 (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. Fig. S4 (d) shows the aggregated attention of the cell decoder when generating the content of each cell. Compared to the structure decoder, the cell decoder has more focused attention, which falls on the cell content that is being generated.

6.4 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. Fig. 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



(a) Input table

(b) Ground truth

(c) EDD (TEDS = 99.8%)

(d) WYGIWYS (TEDS = 89.8%)

(e) Acrobat® on PDF (TEDS = 74.8%)

(f) Acrobat® on Image (TEDS = 64.2%)

(g) Tabula (TEDS = 47.5%)

(h) Traprange (TEDS = 40.2%)

(i) Camelot (TEDS = 35.5%)

(j) PDFPlumber (TEDS = 30.0%)

Fig. 4: 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.

models with tables that satisfy Equation 3, where the thresholds are indicated with vertical dashed lines in Fig. 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 [2] and using different rescaling sizes for each group.

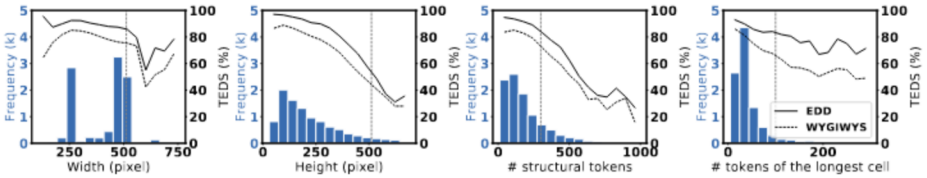


Fig. 5: Impact of table size in terms of width, height, number of structural tokens, and number of tokens in the longest cell on the performance of EDD and WYGIWYS. The bar plots (left axis) are the histogram of PubTabNet test set w.r.t. the above properties. The line plots (right axis) are the mean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

6.5 Generalization

To demonstrate that the EDD model is not only suitable for PubTabNet, but also generalizable to other table recognition datasets, we train and test EDD on the synthetic dataset proposed in [26]. We did not choose the ICDAR2013 or ICDAR2019 table recognition competition datasets. Because, as shown in Table 1, ICDAR2013 does not provide enough training data; and ICDAR2019 does not provide ground truth of cell content (cell position only). We synthesize 500K table images with the corresponding HTML representation[15], evenly distributed among the four categories of table styles defined in [26] (see Fig. S3 in supplemental material for example). The synthetic data is partitioned (stratified sampling by category) into 420K/40k/40k training/validation/test sets.

We compare the test performance of EDD to the graph neural network model TIES proposed in [26] on each table category. We compute the TEDS score only for EDD, as TIES predicts if two tokens (recognized by an OCR engine from the table image) share the same cell, row, and column, but not a HTML representation of the table[16]. Instead, as in [26], the exact match percentage is calculated and compared between EDD and TIES. Note that the exact match for TIES only checks if the cell, row, and column adjacency matrices of the tokens perfectly match the ground truth, but does not check if the OCR engine makes any mistakes. For a fair comparison, we also ignore cell content recognition errors when checking the exact match for EDD, i.e., the recognized table is considered as an exact match as long as the structure perfectly matches the ground truth.

Table 3 shows the test performance of EDD and TIES, where EDD achieves an extremely high TEDS score (99.7+%) on all the categories of the synthetic dataset. This means EDD is able to nearly perfectly reconstructed both the structure and cell content from the table images. EDD outperforms TIES in terms of exact match on all table categories. In addition, unlike TIES, EDD does not show any significant downgrade in performance on category 3 or 4, in

[15] https://github.com/hassan-mahmood/TIES_DataGeneration

[16] [26] does not describe how the adjacency relations can be converted to a unique HTML representation.

which the samples have a more complex structure. This demonstrates that EDD is much more robust and generalizable than TIES on more difficult examples.

Model	Average TEDS (%)				Exact match (%)			
	C1	C2	C3	C4	C1	C2	C3	C4
TIES	–	–	–	–	96.9	94.7	52.9	68.5
EDD	99.8	99.8	99.8	99.7	99.7	99.9	97.2	98.0

Table 3: Test performance of EDD and TIES on the dataset proposed in [26]. TEDS score is not computed for TIES, as it does not generate the HTML representation of input image.

7 Conclusion

This paper makes a comprehensive study of the image-based table recognition problem. A large-scale dataset PubTabNet is developed to train and evaluate deep learning models. 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. Compare to the traditional adjacency relation metric, TEDS can more appropriately capture multi-hop cell misalignment and OCR errors. The proposed EDD model, when trained on PubTabNet, 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. Cesarini, F., Marinai, S., Sarti, L., Soda, G.: Trainable table location in document images. In: Object recognition supported by user interaction for service robots. vol. 3, pp. 236–240. IEEE (2002)
2. Deng, Y., Kanervisto, A., Ling, J., Rush, A.M.: Image-to-markup generation with coarse-to-fine attention. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 980–989. JMLR. org (2017)
3. Deng, Y., Rosenberg, D., Mann, G.: Challenges in end-to-end neural scientific table recognition. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 894–901. IEEE (Sep 2019). https://doi.org/10.1109/ICDAR.2019.00166
4. Fan, M., Kim, D.S.: Table region detection on large-scale pdf files without labeled data. CoRR, abs/1506.08891 (2015)
5. Fang, J., Tao, X., Tang, Z., Qiu, R., Liu, Y.: Dataset, ground-truth and performance metrics for table detection evaluation. In: 2012 10th IAPR International Workshop on Document Analysis Systems. pp. 445–449. IEEE (2012)
6. Gao, L., Huang, Y., Li, Y., Yan, Q., Fang, Y., Dejean, H., Kleber, F., Lang, E.M.: ICDAR 2019 competition on table detection and recognition. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 1510–1515. IEEE (Sep 2019). https://doi.org/10.1109/ICDAR.2019.00166
7. Gatos, B., Danatsas, D., Pratikakis, I., Perantonis, S.J.: Automatic table detection in document images. In: International Conference on Pattern Recognition and Image Analysis. pp. 609–618. Springer (2005)
8. Gilani, A., Qasim, S.R., Malik, I., Shafait, F.: Table detection using deep learning. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). vol. 1, pp. 771–776. IEEE (2017)
9. Göbel, M., Hassan, T., Oro, E., Orsi, G.: ICDAR 2013 table competition. In: 2013 12th International Conference on Document Analysis and Recognition. pp. 1449–1453. IEEE (2013)
10. Hao, L., Gao, L., Yi, X., Tang, Z.: A table detection method for pdf documents based on convolutional neural networks. In: 2016 12th IAPR Workshop on Document Analysis Systems (DAS). pp. 287–292. IEEE (2016)
11. He, D., Cohen, S., Price, B., Kifer, D., Giles, C.L.: 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, pp. 254–261. IEEE (2017)
12. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
14. Hirayama, Y.: A method for table structure analysis using dp matching. In: Proceedings of 3rd International Conference on Document Analysis and Recognition. vol. 2, pp. 583–586. IEEE (1995)
15. Hu, J., Kashi, R.S., Lopresti, D.P., Wilfong, G.: Medium-independent table detection. In: Document Recognition and Retrieval VII. vol. 3967, pp. 291–302. International Society for Optics and Photonics (1999)
16. Hurst, M.: A constraint-based approach to table structure derivation (2003)

17. Jimeno Yepes, A., Verspoor, K.: Literature mining of genetic variants for curation: quantifying the importance of supplementary material. Database **2014** (2014)
18. Kasar, T., Barlas, P., Adam, S., Chatelain, C., Paquet, T.: Learning to detect tables in scanned document images using line information. In: 2013 12th International Conference on Document Analysis and Recognition. pp. 1185–1189. IEEE (2013)
19. Kavasidis, I., Pino, C., Palazzo, S., Rundo, F., Giordano, D., Messina, P., Spampinato, C.: A saliency-based convolutional neural network for table and chart detection in digitized documents. In: International Conference on Image Analysis and Processing. pp. 292–302. Springer (2019)
20. Kieninger, T., Dengel, A.: The t-recs table recognition and analysis system. In: International Workshop on Document Analysis Systems. pp. 255–270. Springer (1998)
21. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Proceedings of International Conference on Learning Representations (ICLR) (2015)
22. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. In: Soviet physics doklady. vol. 10, pp. 707–710 (1966)
23. Morais, R., Le, V., Tran, T., Saha, B., Mansour, M., Venkatesh, S.: Learning regularity in skeleton trajectories for anomaly detection in videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 11996–12004 (2019)
24. Paliwal, S.S., Vishwanath, D., Rahul, R., Sharma, M., Vig, L.: Tablenet: Deep learning model for end-to-end table detection and tabular data extraction from scanned document images. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 128–133. IEEE (2019)
25. Pawlik, M., Augsten, N.: Tree edit distance: Robust and memory-efficient. Information Systems **56**, 157–173 (2016)
26. Qasim, S.R., Mahmood, H., Shafait, F.: Rethinking table recognition using graph neural networks pp. 142–147 (Sep 2019). https://doi.org/10.1109/ICDAR.2019.00166
27. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)
28. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems. pp. 91–99 (2015)
29. Riba, P., Dutta, A., Goldmann, L., Fornés, A., Ramos, O., Lladós, J.: Table detection in invoice documents by graph neural networks. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 122–127. IEEE (Sep 2019). https://doi.org/10.1109/ICDAR.2019.00028
30. Schreiber, S., Agne, S., Wolf, I., Dengel, A., Ahmed, S.: 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, pp. 1162–1167. IEEE (2017)
31. Shafait, F., Smith, R.: Table detection in heterogeneous documents. In: Proceedings of the 9th IAPR International Workshop on Document Analysis Systems. pp. 65–72. ACM (2010)
32. Shahab, A., Shafait, F., Kieninger, T., Dengel, A.: An open approach towards the benchmarking of table structure recognition systems. In: Proceedings of the 9th IAPR International Workshop on Document Analysis Systems. pp. 113–120. ACM (2010)

33. Siegel, N., Lourie, N., Power, R., Ammar, W.: Extracting scientific figures with distantly supervised neural networks. In: Proceedings of the 18th ACM/IEEE on joint conference on digital libraries. pp. 223–232. ACM (2018)
34. e Silva, A.C.: Learning rich hidden markov models in document analysis: Table location. In: 2009 10th International Conference on Document Analysis and Recognition. pp. 843–847. IEEE (2009)
35. Staar, P.W., Dolfi, M., Auer, C., Bekas, C.: 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. pp. 774–782. ACM (2018)
36. Tensmeyer, C., Morariu, V.I., Price, B., Cohen, S., Martinez, T.: Deep splitting and merging for table structure decomposition. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 114–121. IEEE (2019)
37. Tupaj, S., Shi, Z., Chang, C.H., Alam, H.: Extracting tabular information from text files. EECS Department, Tufts University, Medford, USA (1996)
38. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. In: International conference on machine learning. pp. 2048–2057 (2015)
39. Zhong, X., Tang, J., Yepes, A.J.: Publaynet: largest dataset ever for document layout analysis. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 1015–1022. IEEE (Sep 2019). https://doi.org/10.1109/ICDAR.2019.00166
40. Zhou, Y.F., Jiang, R.H., Wu, X., He, J.Y., Weng, S., Peng, Q.: Branchgan: Unsupervised mutual image-to-image transfer with a single encoder and dual decoders. IEEE Transactions on Multimedia (2019)