TableFormer: Table Structure Understanding with Transformers.

Ahmed Nassar, Nikolaos Livathinos, Maksym Lysak, Peter Staar IBM Research

{ahn,nli,mly,taa}@zurich.ibm.com

Abstract

Tables organize valuable content in a concise and compact representation. This content is extremely valuable for systems such as search engines, Knowledge Graph's, etc, since they enhance their predictive capabilities. Unfortunately, tables come in a large variety of shapes and sizes. Furthermore, they can have complex column/row-header configurations, multiline rows, different variety of separation lines, missing entries, etc. As such, the correct identification of the table-structure from an image is a nontrivial task. In this paper, we present a new table-structure identification model. The latter improves the latest end-toend deep learning model (i.e. encoder-dual-decoder from PubTabNet) in two significant ways. First, we introduce a new object detection decoder for table-cells. In this way, we can obtain the content of the table-cells from programmatic PDF's directly from the PDF source and avoid the training of the custom OCR decoders. This architectural change leads to more accurate table-content extraction and allows us to tackle non-english tables. Second, we replace the LSTM decoders with transformer based decoders. This upgrade improves significantly the previous state-of-the-art tree-editing-distance-score (TEDS) from 91% to 98.5% on simple tables and from 88.7% to 95% on complex tables.

1. Introduction

The occurrence of tables in documents is ubiquitous. They often summarise quantitative or factual data, which is cumbersome to describe in verbose text but nevertheless extremely valuable. Unfortunately, this compact representation is often not easy to parse by machines. There are many implicit conventions used to obtain a compact table representation. For example, tables often have complex columnand row-headers in order to reduce duplicated cell content. Lines of different shapes and sizes are leveraged to separate content or indicate a tree structure. Additionally, tables can also have empty/missing table-entries or multi-row textual table-entries. Fig. 1 shows a table which presents all these issues.

a. Picture of a table:

			Observer	1
	6	benign	malignant	Total observer 2
2	benign	13	2	15
Observer 2	malignant	0	62	62
	Total observer 1	13	64	77

b. Red-annotation of bounding boxes, Blue-predictions by TableFormer



c. Structure predicted by TableFormer:

0		1	2	0	
3		4 3	5	6	7
		9	10	11	12
8	2	13	14	15	16
		17	18	19	20

Figure 1: Picture of a table with subtle, complex features such as (1) multi-column headers, (2) cell with multi-row text and (3) cells with no content. Image from PubTabNet evaluation set, filename: 'PMC2944238_004_02'.

Recently, significant progress has been made with vision based approaches to extract tables in documents. For the sake of completeness, the issue of table extraction from documents is typically decomposed into two separate challenges, i.e. (1) finding the location of the table(s) on a document-page and (2) finding the structure of a given table in the document.

The first problem is called table-location and has been previously addressed [30, 38, 19, 21, 23, 26, 8] with stateof-the-art object-detection networks (e.g. YOLO and later on Mask-RCNN [9]). For all practical purposes, it can be considered as a solved problem, given enough ground-truth data to train on.

The second problem is called table-structure decomposition. The latter is a long standing problem in the community of document understanding [6, 4, 14]. Contrary to the table-location problem, there are no commonly used approaches that can easily be re-purposed to solve this problem. Lately, a set of new model-architectures has been proposed by the community to address table-structure decomposition [37, 36, 18, 20]. All these models have some weaknesses (see Sec. 2). The common denominator here is the reliance on textual features and/or the inability to provide the bounding box of each table-cell in the original image.

In this paper, we want to address these weaknesses and present a robust table-structure decomposition algorithm. The design criteria for our model are the following. First, we want our algorithm to be language agnostic. In this way, we can obtain the structure of any table, irregardless of the language. Second, we want our algorithm to leverage as much data as possible from the original PDF document. For programmatic PDF documents, the text-cells can often be extracted much faster and with higher accuracy compared to OCR methods. Last but not least, we want to have a direct link between the table-cell and its bounding box in the image.

To meet the design criteria listed above, we developed a new model called **TableFormer** and a synthetically generated table structure dataset called **SynthTabNet**¹. In particular, our contributions in this work can be summarised as follows:

- We propose **TableFormer**, a transformer based model that predicts tables structure and bounding boxes for the table content simultaneously in an end-to-end approach.
- Across all benchmark datasets TableFormer significantly outperforms existing state-of-the-art metrics, while being much more efficient in training and inference to existing works.
- We present **SynthTabNet** a synthetically generated dataset, with various appearance styles and complexity.
- An augmented dataset based on PubTabNet [37], FinTabNet [36], and TableBank [17] with generated ground-truth for reproducibility.

The paper is structured as follows. In Sec. 2, we give a brief overview of the current state-of-the-art. In Sec. 3, we describe the datasets on which we train. In Sec. 4, we introduce the TableFormer model-architecture and describe its results & performance in Sec. 5. As a conclusion, we describe how this new model-architecture can be re-purposed for other tasks in the computer-vision community.

2. Previous work and State of the Art

Identifying the structure of a table has been an outstanding problem in the document-parsing community, that motivates many organised public challenges [6, 4, 14]. The difficulty of the problem can be attributed to a number of factors. First, there is a large variety in the shapes and sizes of tables. Such large variety requires a flexible method. This is especially true for complex column- and row headers, which can be extremely intricate and demanding. A second factor of complexity is the lack of data with regard to table-structure. Until the publication of PubTabNet [37], there were no large datasets (i.e. > 100K tables) that provided structure information. This happens primarily due to the fact that tables are notoriously time-consuming to annotate by hand. However, this has definitely changed in recent years with the deliverance of PubTabNet [37], FinTab-Net [36], TableBank [17] etc.

Before the rising popularity of deep neural networks, the community relied heavily on heuristic and/or statistical methods to do table structure identification [3, 7, 11, 5, 13, 28]. Although such methods work well on constrained tables [12], a more data-driven approach can be applied due to the advent of convolutional neural networks (CNNs) and the availability of large datasets. To the best-of-our knowledge, there are currently two different types of network architecture that are being pursued for state-of-the-art tablestructure identification.

Image-to-Text networks: In this type of network, one predicts a sequence of tokens starting from an encoded image. Such sequences of tokens can be HTML table tags [37, 17] or LaTeX symbols[10]. The choice of symbols is ultimately not very important, since one can be transformed into the other. There are however subtle variations in the Image-to-Text networks. The easiest network architectures are "image-encoder \rightarrow text-decoder" (IETD), similar to network architectures that try to provide captions to images [32]. In these IETD networks, one expects as output the LaTeX/HTML string of the entire table, i.e. the symbols necessary for creating the table with the content of the table. Another approach is the "image-encoder \rightarrow dual decoder" (IEDD) networks. In these type of networks, one has two consecutive decoders with different purposes. The first decoder is the *tag-decoder*, i.e. it only produces the HTM-L/LaTeX tags which construct an empty table. The second content-decoder uses the encoding of the image in combination with the output encoding of each cell-tag (from the tag-decoder) to generate the textual content of each table cell. The network architecture of IEDD is certainly more elaborate, but it has the advantage that one can pre-train the

¹https://github.com/IBM/SynthTabNet

tag-decoder which is constrained to the table-tags.

In practice, both network architectures (IETD and IEDD) require an implicit, custom trained object-characterrecognition (OCR) to obtain the content of the table-cells. In the case of IETD, this OCR engine is implicit in the decoder similar to [24]. For the IEDD, the OCR is solely embedded in the content-decoder. This reliance on a custom. implicit OCR decoder is of course problematic. OCR is a well known and extremely tough problem, that often needs custom training for each individual language. However, the limited availability for non-english content in the current datasets, makes it impractical to apply the IETD and IEDD methods on tables with other languages. Additionally, OCR can be completely omitted if the tables originate from programmatic PDF documents with known positions of each cell. The latter was the inspiration for the work of this paper.

Graph Neural networks: Graph Neural networks (GNN's) take a radically different approach to tablestructure extraction. Note that one table cell can constitute out of multiple text-cells. To obtain the table-structure, one creates an initial graph, where each of the text-cells becomes a node in the graph similar to [33, 34, 2]. Each node is then associated with en embedding vector coming from the encoded image, its coordinates and the encoded text. Furthermore, nodes that represent adjacent text-cells are linked. Graph Convolutional Networks (GCN's) based methods take the image as an input, but also the position of the text-cells and their content [18]. The purpose of a GCN is to transform the input graph into a new graph, which replaces the old links with new ones. The new links then represent the table-structure. With this approach, one can avoid the need to build custom OCR decoders. However, the quality of the reconstructed structure is not comparable to the current state-of-the-art [18].

Hybrid Deep Learning-Rule-Based approach: A popular current model for table-structure identification is the use of a hybrid Deep Learning-Rule-Based approach similar to [27, 29]. In this approach, one first detects the position of the table-cells with object detection (e.g. YoloVx or Mask-RCNN), then classifies the table into different types (from its images) and finally uses different rule-sets to obtain its table-structure. Currently, this approach achieves stateof-the-art results, but is not an end-to-end deep-learning method. As such, new rules need to be written if different types of tables are encountered.

3. Datasets

We rely on large-scale datasets such as PubTabNet [37], FinTabNet [36], and TableBank [17] datasets to train and evaluate our models. These datasets span over various appearance styles and content. We also introduce our own synthetically generated SynthTabNet dataset to fix an im-



Figure 2: Distribution of the tables across different table dimensions in PubTabNet + FinTabNet datasets

balance in the previous datasets.

The PubTabNet dataset contains 509k tables delivered as annotated PNG images. The annotations consist of the table structure represented in HTML format, the tokenized text and its bounding boxes per table cell. Fig. 1 shows the appearance style of PubTabNet. Depending on its complexity, a table is characterized as "simple" when it does not contain row spans or column spans, otherwise it is "complex". The dataset is divided into Train and Val splits (roughly 98% and 2%). The Train split consists of 54% simple and 46% complex tables and the Val split of 51% and 49% respectively. The FinTabNet dataset contains 112k tables delivered as single-page PDF documents with mixed table structures and text content. Similarly to the PubTabNet, the annotations of FinTabNet include the table structure in HTML, the tokenized text and the bounding boxes on a table cell basis. The dataset is divided into Train, Test and Val splits (81%, 9.5%, 9.5%), and each one is almost equally divided into simple and complex tables (Train: 48% simple, 52% complex, Test: 48% simple, 52% complex, Test: 53% simple, 47% complex). Finally the TableBank dataset consists of 145k tables provided as JPEG images. The latter has annotations for the table structure, but only few with bounding boxes of the table cells. The entire dataset consists of simple tables and it is divided into 90% Train, 3% Test and 7% Val splits.

Due to the heterogeneity across the dataset formats, it was necessary to combine all available data into one homogenized dataset before we could train our models for practical purposes. Given the size of PubTabNet, we adopted its annotation format and we extracted and converted all tables as PNG images with a resolution of 72 dpi. Additionally, we have filtered out tables with extreme sizes due to small amount of such tables, and kept only those ones ranging between 1*1 and 20*10 (rows/columns).

The availability of the bounding boxes for all table cells is essential to train our models. In order to distinguish between empty and non-empty bounding boxes, we have introduced a binary class in the annotation. Unfortunately, the original datasets either omit the bounding boxes for whole tables (e.g. TableBank) or they narrow their scope only to non-empty cells. Therefore, it was imperative to introduce a data pre-processing procedure that generates the missing bounding boxes out of the annotation information. This procedure first parses the provided table structure and calculates the dimensions of the most fine-grained grid that covers the table structure. Notice that each table cell may occupy multiple grid squares due to row or column spans. In case of PubTabNet we had to compute missing bounding boxes for 48% of the simple and 69% of the complex tables. Regarding FinTabNet, 68% of the simple and 98% of the complex tables require the generation of bounding boxes.

As it is illustrated in Fig. 2, the table distributions from all datasets are skewed towards simpler structures with fewer number of rows/columns. Additionally, there is very limited variance in the table styles, which in case of Pub-TabNet and FinTabNet means one styling format for the majority of the tables. Similar limitations appear also in the type of table content, which in some cases (e.g. FinTab-Net) is restricted to a certain domain. Ultimately, the lack of diversity in the training dataset damages the ability of the models to generalize well on unseen data.

Motivated by those observations we aimed at generating a synthetic table dataset named SynthTabNet. This approach offers control over: 1) the size of the dataset, 2) the table structure, 3) the table style and 4) the type of content. The complexity of the table structure is described by the size of the table header and the table body, as well as the percentage of the table cells covered by row spans and column spans. A set of carefully designed styling templates provides the basis to build a wide range of table appearances. Lastly, the table content is generated out of a curated collection of text corpora. By controlling the size and scope of the synthetic datasets we are able to train and evaluate our models in a variety of different conditions. For example, we can first generate a highly diverse dataset to train our models and then evaluate their performance on other synthetic datasets which are focused on a specific domain.

In this regard, we have prepared four synthetic datasets, each one containing 150k examples. The corpora to generate the table text consists of the most frequent terms appearing in PubTabNet and FinTabNet together with randomly generated text. The first two synthetic datasets have been fine-tuned to mimic the appearance of the original datasets but encompass more complicated table structures. The third

	Tags	Bbox	Size	Format
PubTabNet	1	1	509k	PNG
FinTabNet	1	1	112k	PDF
TableBank	1	×	145k	JPEG
Combined-Tabnet(*)	1	1	400k	PNG
Combined(**)	1	1	500k	PNG
SynthTabNet	1	1	600k	PNG

Table 1: Both "*Combined-Tabnet*" and "*Combined-Tabnet*" are variations of the following: (*) The Combined-Tabnet dataset is the processed combination of PubTabNet and Fintabnet. (**) The combined dataset is the processed combination of PubTabNet, Fintabnet and TableBank.

one adopts a colorful appearance with high contrast and the last one contains tables with sparse content. Lastly, we have combined all synthetic datasets into one big unified synthetic dataset of 600k examples.

Tab. 1 summarizes the various attributes of the datasets.

4. The TableFormer model

Given the image of a table, TableFormer is able to predict: 1) a sequence of tokens that represent the structure of a table, and 2) a bounding box coupled to a subset of those tokens. The conversion of an image into a sequence of tokens is a well-known task [35, 16]. While attention is often used as an implicit method to associate each token of the sequence with a position in the original image, an explicit association between the individual table-cells and the image bounding boxes is also required.

4.1. Model architecture.

We now describe in detail the proposed method, which is composed of three main components, see Fig. 4. Our CNN Backbone Network encodes the input as a feature vector of predefined length. The input feature vector of the encoded image is passed to the Structure Decoder to produce a sequence of HTML tags that represent the structure of the table. With each prediction of an HTML standard data cell ('') the hidden state of that cell is passed to the Cell BBox Decoder. As for spanning cells, such as row or column span, the tag is broken down to '<', 'rowspan=' or 'colspan=', with the number of spanning cells (attribute), and '>'. The hidden state attached to '<' is passed to the Cell BBox Decoder. A shared feed forward network (FFN) receives the hidden states from the Structure Decoder, to provide the final detection predictions of the bounding box coordinates and their classification.

CNN Backbone Network. A ResNet-18 CNN is the backbone that receives the table image and encodes it as a vector of predefined length. The network has been modified by removing the linear and pooling layer, as we are not per-



Figure 3: **TableFormer** takes in an image of the PDF and creates bounding box and HTML structure predictions that are synchronized. The bounding boxes grabs the content from the PDF and inserts it in the structure.



Figure 4: Given an input image of a table, the **Encoder** produces fixed-length features that represent the input image. The features are then passed to both the **Structure Decoder** and **Cell BBox Decoder**. During training, the **Structure Decoder** receives 'tokenized tags' of the HTML code that represent the table structure. Afterwards, a transformer encoder and decoder architecture is employed to produce features that are received by a linear layer, and the **Cell BBox Decoder** selects features referring to the data cells ('', '<') and passes them through an attention network, an MLP, and a linear layer to predict the bounding boxes.

forming classification, and adding an adaptive pooling layer of size 28*28. ResNet by default downsamples the image resolution by 32 and then the encoded image is provided to both the *Structure Decoder*, and *Cell BBox Decoder*.

Structure Decoder. The transformer architecture of this component is based on the work proposed in [31]. After extensive experimentation, the *Structure Decoder* is modeled as a transformer encoder with two encoder layers and a transformer decoder made from a stack of 4 decoder layers that comprise mainly of multi-head attention and feed forward layers. This configuration uses fewer layers and heads in comparison to networks applied to other problems (e.g. "Scene Understanding", "Image Captioning"), something which we relate to the simplicity of table images.

The transformer encoder receives an encoded image from the *CNN Backbone Network* and refines it through a multi-head dot-product attention layer, followed by a Feed Forward Network. During training, the transformer decoder receives as input the output feature produced by the transformer encoder, and the tokenized input of the HTML ground-truth tags. Using a stack of multi-head attention layers, different aspects of the tag sequence could be inferred. This is achieved by each attention head on a layer operating in a different subspace, and then combining altogether their attention score.

Cell BBox Decoder. Our architecture allows to simultaneously predict HTML tags and bounding boxes for each table cell without the need of a separate object detector end to end. This approach is inspired by DETR [1] which employs a Transformer Encoder, and Decoder that looks for a specific number of object queries (potential object detections). As our model utilizes a transformer architecture, the hidden state of the ' and '<' HTML structure tags become the object query.

The encoding generated by the *CNN Backbone Network* along with the features acquired for every data cell from the Transformer Decoder are then passed to the attention network. The attention network takes both inputs and learns to provide an attention weighted encoding. This weighted attention encoding is then multiplied to the encoded image to produce a feature for each table cell. Notice that this is different than the typical object detection problem where imbalances between the number of detections and the amount of objects may exist. In our case, we know up front that the produced detections always match with the table cells in number and correspondence.

The output features for each table cell are then fed into the feed-forward network (FFN). The FFN consists of a Multi-Layer Perceptron (3 layers with ReLU activation function) that predicts the normalized coordinates for the bounding box of each table cell. Finally, the predicted bounding boxes are classified based on whether they are empty or not using a linear layer.

Loss Functions. We formulate a multi-task loss Eq. 2 to train our network. The Cross-Entropy loss (denoted as l_s) is used to train the *Structure Decoder* which predicts the structure tokens. As for the Cell BBox Decoder it is trained with a combination of losses denoted as l_{box} . l_{box} consists of the generally used l_1 loss for object detection and the IoU loss (l_{iou}) to be scale invariant as explained in [25]. In comparison to DETR, we do not use the Hungarian algorithm [15] to match the predicted bounding boxes with the ground-truth boxes, as we have already achieved a one-toone match through two steps: 1) Our token input sequence is naturally ordered, therefore the hidden states of the table data cells are also in order when they are provided as input to the Cell BBox Decoder, and 2) Our bounding boxes generation mechanism (see Sec. 3) ensures a one-to-one mapping between the cell content and its bounding box for all post-processed datasets.

The loss used to train the TableFormer can be defined as following:

$$l_{box} = \lambda_{iou} l_{iou} + \lambda_{l1}$$

$$l = \lambda l_s + (1 - \lambda) l_{box}$$
(1)

where $\lambda \in [0, 1]$, and $\lambda_{iou}, \lambda_{l1} \in \mathbb{R}$ are hyper-parameters.

5. Experimental Results

5.1. Implementation Details

TableFormer uses ResNet-18 as the *CNN Backbone Net-work*. The input images are resized to 448*448 pixels and the feature map has a dimension of 28*28. Additionally, we enforce the following input constraints:

Image width and height
$$\leq 1024$$
 pixels(2)Structural tags length ≤ 512 tokens.

Although input constraints are used also by other methods, such as EDD, ours are less restrictive due to the improved

runtime performance and lower memory footprint of Table-Former. This allows to utilize input samples with longer sequences and images with larger dimensions.

The Transformer Encoder consists of two "Transformer Encoder Layers", with an input feature size of 512, feed forward network of 1024, and 4 attention heads. As for the Transformer Decoder it is composed of four "Transformer Decoder Layers" with similar input and output dimensions as the "Transformer Encoder Layers". Even though our model uses fewer layers and heads than the default implementation parameters, our extensive experimentation has proved this setup to be more suitable for table images. We attribute this finding to the inherent design of table images, which contain mostly lines and text, unlike the more elaborate content present in other scopes (e.g. the COCO dataset). Moreover, we have added ResNet blocks to the inputs of the Structure Decoder and Cell BBox Decoder. This prevents a decoder having a stronger influence over the learned weights which would damage the other prediction task (structure vs bounding boxes), but learn task specific weights instead. Lastly our dropout layers are set to 0.5.

For training, TableFormer is trained with 3 Adam optimizers, each one for the *CNN Backbone Network*, *Structure Decoder*, and *Cell BBox Decoder*. Taking the PubTabNet as an example for our parameter set up, the initializing learning rate is 0.001 for 12 epochs with a batch size of 24, and λ set to 0.5. Afterwards, we reduce the learning rate to 0.0001, the batch size to 18 and train for 12 more epochs or convergence.

TableFormer is implemented with PyTorch and Torchvision libraries [22]. To speed up the inference, the image undergoes a single forward pass through the *CNN Backbone Network* and transformer encoder. This eliminates the overhead of generating the same features for each decoding step. Similarly, we employ a 'caching' technique to preform faster autoregressive decoding. This is achieved by storing the features of decoded tokens so we can reuse them for each time step. Therefore, we only compute the attention for each new tag.

5.2. Generalization

TableFormer is evaluated on three major publicly available datasets of different nature to prove the generalization and effectiveness of our model. The datasets used for evaluation are the PubTabNet, FinTabNet and TableBank which stem from the scientific, financial and general domains respectively.

We also share our baseline results on the challenging SynthTabNet dataset. Throughout our experiments, the same parameters stated in Sec. 5.1 are utilized.

5.3. Datasets and Metrics

The Tree-Edit-Distance-Based Similarity (TEDS) metric was introduced in [37]. It represents the prediction, and ground-truth as a tree structure of HTML tags. This similarity is calculated as:

$$\operatorname{TEDS}\left(T_{a}, T_{b}\right) = 1 - \frac{\operatorname{EditDist}\left(T_{a}, T_{b}\right)}{\max\left(\left|T_{a}\right|, \left|T_{b}\right|\right)}$$
(3)

where T_a and T_b represent tables in tree structure HTML format. EditDist denotes the tree-edit distance, and |T| represents the number of nodes in T.

5.4. Quantitative Analysis

Structure. As shown in Tab. 2, TableFormer outperforms all SOTA methods across different datasets by a large margin for predicting the table structure from an image. All the more, our model outperforms pre-trained methods. During the evaluation we do not apply any table filtering. We also provide our baseline results on the SynthTabNet dataset. It has been observed that large tables (e.g. tables that occupy half of the page or more) yield poor predictions. We attribute this issue to the image resizing during the pre-processing step, that produces downsampled images with indistinguishable features. This problem can be addressed by treating such big tables with a separate model which accepts a large input image size.

Madal			TEDS	
Model	Dataset	Simple	Complex	All
EDD	PTN	91.1	88.7	89.9
GTE	PTN	-	-	93.01
TableFormer	PTN	98.5	95.0	96.75
EDD	FTN	88.4	92.08	90.6
GTE	FTN	-	-	87.14
GTE (FT)	FTN	-	-	91.02
TableFormer	FTN	97.5	96.0	96.8
EDD	TB	86.0	-	86.0
TableFormer	TB	89.6	-	89.6
TableFormer	STN	96.9	95.7	96.7

Table 2: Structure results on PubTabNet (PTN), FinTabNet (FTN), TableBank (TB) and SynthTabNet (STN). FT: Model was trained on PubTabNet then finetuned.

Cell Detection. Like any object detector, our *Cell BBox Detector* provides bounding boxes that can be improved with post-processing during inference. We make use of the grid-like structure of tables to refine the predictions. A detailed explanation on the post-processing is available in the supplementary material. As shown in Tab. 3, we evaluate our *Cell BBox Decoder* accuracy for cells with a class label of 'content' only using the PASCAL VOC mAP metric for pre-processing and post-processing. Note that we do not have post-processing results for SynthTabNet as images are only provided. To compare the performance of our proposed approach, we've integrated TableFormer's *Cell BBox Decoder* into EDD architecture. As mentioned previously, the Structure Decoder provides the *Cell BBox Decoder* with the features needed to predict the bounding box predictions. Therefore, the accuracy of the *Structure Decoder*. If the *Structure Decoder* predicts an extra column, this will result in an extra column of predicted bounding boxes.

Model	Dataset	mAP	mAP (PP)
EDD+BBox	PubTabNet	79.2	82.7
TableFormer	PubTabNet	82.1	86.8
TableFormer	SynthTabNet	87.7	-

Table 3: Cell Bounding Box detection results on PubTab-Net, and FinTabNet. PP: Post-processing.

Cell Content. In this section, we evaluate the entire pipeline of recovering a table with content. Here we put our approach to test by capitalizing on extracting content from the PDF cells rather than decoding from images. Tab. 4 shows the TEDs score of HTML code representing the structure of the table along with the content inserted in the data cell and compared with the ground-truth. Our method achieved a **5.3%** increase over the state-of-the-art, and commercial solutions. We believe our scores would be higher if the HTML ground-truth matched the extracted PDF cell content. Unfortunately, there are small discrepancies such as spacings around words or special characters with various unicode representations.

Simple	TEDS Complex	All
78.0	57.8	67.9
60.8	49.9	55.4
80.0	66.0	73.0
68.9	61.8	65.3
91.2	85.4	88.3
95.4	90.1	93.6
	Simple 78.0 60.8 80.0 68.9 91.2 95.4	TEDSSimpleComplex78.057.860.849.980.066.068.961.891.285.495.490.1

Table 4: Results of structure with content retrieved using cell detection on PubTabNet. In all cases the input is PDF documents with cropped tables.

a. Red - PDF cells, Green - predicted bounding boxes, Blue - post-processed predictions matched to PDF cells

Japanese language (previously unseen by TableFormer):

,		.,			/-	
		論文ス	アイル	参考	文献	
	ファイル数	央語	日本語	央語	日本語	
ics(ACL2003)	65	65	U	150	0	
unal .		1.210		1150		

Example table from FinTabNet:

		調义。	ノアイル	変す	与又厥		lin mill	ione)	Crant Data	Eair Value			
出典	ファイル数	央記	日本語	央語	日本語		un min	ions	Grant Date	Fair value			
Association for Computational Linguistics (ACL2003)	<u>68</u>	65	U	150	U		RSUs	PSUs	RSUs	PSUs			
Computational Linguistics(COLING2002)	140	140		150	U	Nonvested on January 1	1.1	0.3	\$ 90.10	\$ 91.1			
電気情報通信学会 2003 年総合大会	150	8	142	225	147	Granted	0.5	0 1	117 44	122.4			
情報処理学会第65回全国大会(2003)	177		176	150	236	Vested	(0.5)	10.11	97.09	01.1			
第17回人工知能学会全国大会 (2003)	208	E	205	152	244	vesieu	(0.3)	(0.1)	φ7.00	01.1			
自然言語処理研究会第146~155回	98	2	<u>916</u>	150	232	Canceled or forfeited	(0.1)		102.01	92.18			
WWWから収集した論文	107	76	52	147	96	Nonvested on December 31	1.0	0.3	\$ 104.85	\$ 104.5			
it is the second s	<u>945</u>	294	651	1122	955				-				
•													
b. Structure predicted by Table	Structure predicted by TableFormer, with superimposed matched PDF cell text:												
	*0.			60 1 7 ->	-+h	Charge	(in	Mainte		A Data Fair			

		調文ノ	111	25			Shares (in i	nares (in minoris) weighted Average Grant Date Fall					
出典	ファイル数	英語	日本語	英語	日本語				Vai	lue			
Association for Computational Linguistics(ACL2003)	65	65	0	150	0		RSUs	PSUs	RSUs	PSUs			
Computational Linguistics(COLING2002)	140	140	0	150	0	Nonvested on January 1	11	0.3	90.10.\$	\$ 91 19			
電気情報通信学会2003年総合大会	150	8	142	223	147		1.1	0.0	00.10 φ	¢01.10			
情報処理学会第65回全国大会(2003)	177	1	176	150	236	Granted	0.5	0.1	117.44	122.41			
第17回人工知能学会全国大会(2003)	208	5	203	152	244	Vested	(0.5)	(0.1)	87.08	81.14			
自然言語処理研究会第146~155回	98	2	96	150	232	Canceled or forfeited	(0.1)	_	102.01	92.18			
WWWから収集した論文	107	73	34	147	96	Nonvested on December 31	1.0	0.3	104.85 \$	\$ 104.51			
計	945	294	651	1122	955	L							

Text is aligned to match original for ease of viewing

Maighted Ave

Figure 5: One of the benefits of TableFormer is that it is language agnostic, as an example, the left part of the illustration demonstrates TableFormer predictions on previously unseen language (Japanese). Additionally, we see that TableFormer is robust to variability in style and content, right side of the illustration shows the example of the TableFormer prediction from the FinTabNet dataset.

Gro	ound Ti	ruth						Red - PDF cells, Green - predicted bounding boxes										Predic	ted S	Struct	ure													
Fourt	h er	December 31,		Basic	Income from continuing operations	Second Quarter	Canada	Net earnings	Year Ended December 31,	Interest income	Net cash provided by operating activities	Fourth Quarter		December 31,		Baste	income from continuin operation	Quarte	Canada	Net Farnings Decemi 31,	er income by operation	sh ed ing	0	1	2	3	4	5	6	7	8	9	10	11
Inter cost	ist Service cost	s-	In	Shares	in millions	High	Discount rate		Other assets	-%	ocurracy.	Interest	Service cost	5	In	Shares	n million	High	Discount	Other	-		12	13	14	1.5	16	17	18	19	20	21	22	
	Total revenues	First Quarter	thousands	Amount	\$ in millions	Year En	ded Decemb	er 31,	\$ in millio	ns			Total	First Quarter	thousands	Amount	s in millions	Near E	ided December	31. Sinni	lions		23	24	25	15	26	27	28			29		
	Revenue		-	Shares	48.86%	% Change	Fourth Quarter	Year Ended December 31,	Low	Net cash provided by operating activities	Interest cost		Revenue		8	Shine	48.86%	% Change	Fourth Quarter	Ended December 11,	Net cash provided by operating activities	I I	30	31	32	33	34	35	36	37	38	39	40	41
Share	5	ş.	in thousands	Year Ended December 31,	8-K	u.s.	Thereafter	s-		Long- term debt	Accounts payable	Shares		•	a theosands	tear Ended December 31.	•		Thereafter	•	Long berm debt		42	43	44	45	46	47	48	49	50	51	52	53

Figure 6: An example of TableFormer predictions (bounding boxes and structure) from generated SynthTabNet table.

5.5. Qualitative Analysis

We showcase several visualizations for the different components of our network on various "complex" tables within datasets presented in this work in Fig. 5 and Fig. 6 As it is shown, our model is able to predict bounding boxes for all table cells, even for the empty ones. Additionally, our post-processing techniques can extract the cell content by matching the predicted bounding boxes to the PDF cells based on their overlap and spatial proximity. The left part of Fig. 5 demonstrates also the adaptability of our method to any language, as it can successfully extract Japanese text, although the training set contains only English content. We provide more visualizations including the intermediate steps in the supplementary material. Overall these illustrations justify the versatility of our method across a diverse range of table appearances and content type.

6. Future Work & Conclusion

In this paper, we presented TableFormer an end-to-end transformer based approach to predict table structures and bounding boxes of cells from an image. This approach enables us to recreate the table structure, and extract the cell content from PDF or OCR by using bounding boxes. Additionally, it provides the versatility required in real-world scenarios when dealing with various types of PDF documents, and languages. Furthermore, our method outperforms all state-of-the-arts with a wide margin. Finally, we introduce "SynthTabNet" a challenging synthetically generated dataset that reinforces missing characteristics from other datasets.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-toend object detection with transformers. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 213–229, Cham, 2020. Springer International Publishing, 5

- [2] Zewen Chi, Heyan Huang, Heng-Da Xu, Houjin Yu, Wanxuan Yin, and Xian-Ling Mao. Complicated table structure recognition. *arXiv preprint arXiv:1908.04729*, 2019. 3
- Bertrand Couasnon and Aurelie Lemaitre. *Recognition of Tables and Forms*, pages 647–677. Springer London, London, 2014. 2
- [4] Hervé Déjean, Jean-Luc Meunier, Liangcai Gao, Yilun Huang, Yu Fang, Florian Kleber, and Eva-Maria Lang. IC-DAR 2019 Competition on Table Detection and Recognition (cTDaR), Apr. 2019. http://sac.founderit.com/. 2
- [5] Basilios Gatos, Dimitrios Danatsas, Ioannis Pratikakis, and Stavros J Perantonis. Automatic table detection in document images. In *International Conference on Pattern Recognition* and Image Analysis, pages 609–618. Springer, 2005. 2
- [6] Max Göbel, Tamir Hassan, Ermelinda Oro, and Giorgio Orsi. Icdar 2013 table competition. In 2013 12th International Conference on Document Analysis and Recognition, pages 1449–1453, 2013. 2
- [7] EA Green and M Krishnamoorthy. Recognition of tables using table grammars. procs. In Symposium on Document Analysis and Recognition (SDAIR'95), pages 261–277. 2
- [8] Khurram Azeem Hashmi, Alain Pagani, Marcus Liwicki, Didier Stricker, and Muhammad Zeshan Afzal. Castabdetectors: Cascade network for table detection in document images with recursive feature pyramid and switchable atrous convolution. *Journal of Imaging*, 7(10), 2021. 1
- [9] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Oct 2017. 1
- [10] Yelin He, X. Qi, Jiaquan Ye, Peng Gao, Yihao Chen, Bingcong Li, Xin Tang, and Rong Xiao. Pingan-vcgroup's solution for icdar 2021 competition on scientific table image recognition to latex. *ArXiv*, abs/2105.01846, 2021. 2
- [11] Jianying Hu, Ramanujan S Kashi, Daniel P Lopresti, and Gordon Wilfong. Medium-independent table detection. In *Document Recognition and Retrieval VII*, volume 3967, pages 291–302. International Society for Optics and Photonics, 1999. 2
- [12] Matthew Hurst. A constraint-based approach to table structure derivation. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition - Volume* 2, ICDAR '03, page 911, USA, 2003. IEEE Computer Society. 2
- [13] Thotreingam Kasar, Philippine Barlas, Sebastien Adam, Clément Chatelain, and Thierry Paquet. Learning to detect tables in scanned document images using line information. In 2013 12th International Conference on Document Analysis and Recognition, pages 1185–1189. IEEE, 2013. 2
- [14] Pratik Kayal, Mrinal Anand, Harsh Desai, and Mayank Singh. Icdar 2021 competition on scientific table image recognition to latex, 2021. 2
- [15] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 6

- [16] Girish Kulkarni, Visruth Premraj, Vicente Ordonez, Sagnik Dhar, Siming Li, Yejin Choi, Alexander C. Berg, and Tamara L. Berg. Babytalk: Understanding and generating simple image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2891–2903, 2013. 4
- [17] Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, Ming Zhou, and Zhoujun Li. Tablebank: A benchmark dataset for table detection and recognition, 2019. 2, 3
- [18] Yiren Li, Zheng Huang, Junchi Yan, Yi Zhou, Fan Ye, and Xianhui Liu. Gfte: Graph-based financial table extraction. In Alberto Del Bimbo, Rita Cucchiara, Stan Sclaroff, Giovanni Maria Farinella, Tao Mei, Marco Bertini, Hugo Jair Escalante, and Roberto Vezzani, editors, *Pattern Recognition. ICPR International Workshops and Challenges*, pages 644–658, Cham, 2021. Springer International Publishing. 2, 3
- [19] Nikolaos Livathinos, Cesar Berrospi, Maksym Lysak, Viktor Kuropiatnyk, Ahmed Nassar, Andre Carvalho, Michele Dolfi, Christoph Auer, Kasper Dinkla, and Peter Staar. Robust pdf document conversion using recurrent neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(17):15137–15145, May 2021. 1
- [20] Rujiao Long, Wen Wang, Nan Xue, Feiyu Gao, Zhibo Yang, Yongpan Wang, and Gui-Song Xia. Parsing table structures in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 944–952, 2021. 2
- [21] Shubham Singh Paliwal, D Vishwanath, Rohit Rahul, Monika Sharma, and Lovekesh Vig. 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), pages 128–133. IEEE, 2019. 1
- [22] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 6
- [23] Devashish Prasad, Ayan Gadpal, Kshitij Kapadni, Manish Visave, and Kavita Sultanpure. Cascadetabnet: An approach for end to end table detection and structure recognition from image-based documents. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 572–573, 2020. 1
- Shah Rukh Qasim, Hassan Mahmood, and Faisal Shafait. Rethinking table recognition using graph neural networks. In 2019 International Conference on Document Analysis and Recognition (ICDAR), pages 142–147. IEEE, 2019. 3
- [25] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF Conference on*

Computer Vision and Pattern Recognition, pages 658–666, 2019. 6

- [26] Sebastian Schreiber, Stefan Agne, Ivo Wolf, Andreas Dengel, and Sheraz 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), volume 01, pages 1162– 1167, 2017. 1
- [27] Sebastian Schreiber, Stefan Agne, Ivo Wolf, Andreas Dengel, and Sheraz 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), volume 1, pages 1162–1167. IEEE, 2017. 3
- [28] Faisal Shafait and Ray Smith. Table detection in heterogeneous documents. In Proceedings of the 9th IAPR International Workshop on Document Analysis Systems, pages 65– 72, 2010. 2
- [29] Shoaib Ahmed Siddiqui, Imran Ali Fateh, Syed Tahseen Raza Rizvi, Andreas Dengel, and Sheraz Ahmed. Deeptabstr: Deep learning based table structure recognition. In 2019 International Conference on Document Analysis and Recognition (ICDAR), pages 1403–1409. IEEE, 2019. 3
- [30] Peter W J Staar, Michele Dolfi, Christoph Auer, and Costas Bekas. Corpus conversion service: A machine learning platform to ingest documents at scale. In *Proceedings of the* 24th ACM SIGKDD, KDD '18, pages 774–782, New York, NY, USA, 2018. ACM. 1
- [31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017. 5
- [32] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015. 2
- [33] Wenyuan Xue, Qingyong Li, and Dacheng Tao. Res2tim: reconstruct syntactic structures from table images. In 2019 International Conference on Document Analysis and Recognition (ICDAR), pages 749–755. IEEE, 2019. 3
- [34] Wenyuan Xue, Baosheng Yu, Wen Wang, Dacheng Tao, and Qingyong Li. Tgrnet: A table graph reconstruction network for table structure recognition. arXiv preprint arXiv:2106.10598, 2021. 3
- [35] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4651–4659, 2016. 4
- [36] Xinyi Zheng, Doug Burdick, Lucian Popa, Peter Zhong, and Nancy Xin Ru Wang. Global table extractor (gte): A framework for joint table identification and cell structure recognition using visual context. *Winter Conference for Applications in Computer Vision (WACV)*, 2021. 2, 3
- [37] Xu Zhong, Elaheh ShafieiBavani, and Antonio Jimeno Yepes. Image-based table recognition: Data, model,

and evaluation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 564–580, Cham, 2020. Springer International Publishing. 2, 3, 7

[38] Xu Zhong, Jianbin Tang, and Antonio Jimeno Yepes. Publaynet: Largest dataset ever for document layout analysis. In 2019 International Conference on Document Analysis and Recognition (ICDAR), pages 1015–1022, 2019. 1

TableFormer: Table Structure Understanding with Transformers Supplementary Material

1. Details on the datasets

1.1. Data preparation

As a first step of our data preparation process, we have calculated statistics over the datasets across the following dimensions: (1) table size measured in the number of rows and columns, (2) complexity of the table, (3) strictness of the provided HTML structure and (4) completeness (i.e. no omitted bounding boxes). A table is considered to be simple if it does not contain row spans or column spans. Additionally, a table has a strict HTML structure if every row has the same number of columns after taking into account any row or column spans. Therefore a strict HTML structure looks always rectangular. However, HTML is a lenient encoding format, i.e. tables with rows of different sizes might still be regarded as correct due to implicit display rules. These implicit rules leave room for ambiguity, which we want to avoid. As such, we prefer to have "strict" tables, i.e. tables where every row has exactly the same length.

We have developed a technique that tries to derive a missing bounding box out of its neighbors. As a first step, we use the annotation data to generate the most fine-grained grid that covers the table structure. In case of strict HTML tables, all grid squares are associated with some table cell and in the presence of table spans a cell extends across multiple grid squares. When enough bounding boxes are known for a rectangular table, it is possible to compute the geometrical border lines between the grid rows and columns. Eventually this information is used to generate the missing bounding boxes. Additionally, the existence of unused grid squares indicates that the table rows have unequal number of columns and the overall structure is non-strict. The generation of missing bounding boxes for non-strict HTML tables is ambiguous and therefore quite challenging. Thus, we have decided to simply discard those tables. In case of PubTabNet we have computed missing bounding boxes for 48% of the simple and 69% of the complex tables. Regarding FinTabNet, 68% of the simple and 98% of the complex tables require the generation of bounding boxes.

Figure 7 illustrates the distribution of the tables across different dimensions per dataset.

1.2. Synthetic datasets

Aiming to train and evaluate our models in a broader spectrum of table data we have synthesized four types of datasets. Each one contains tables with different appearances in regard to their size, structure, style and content. Every synthetic dataset contains 150k examples, summing up to 600k synthetic examples. All datasets are divided into Train, Test and Val splits (80%, 10%, 10%).

The process of generating a synthetic dataset can be decomposed into the following steps:

1. Prepare styling and content templates: The styling templates have been manually designed and organized into groups of scope specific appearances (e.g. financial data, marketing data, etc.) Additionally, we have prepared curated collections of content templates by extracting the most frequently used terms out of non-synthetic datasets (e.g. PubTabNet, FinTabNet, etc.).

2. Generate table structures: The structure of each synthetic dataset assumes a horizontal table header which potentially spans over multiple rows and a table body that may contain a combination of row spans and column spans. However, spans are not allowed to cross the header - body boundary. The table structure is described by the parameters: Total number of table rows and columns, number of header rows, type of spans (header only spans, row only spans, column only spans, both row and column spans), maximum span size and the ratio of the table area covered by spans.

3. Generate content: Based on the dataset *theme*, a set of suitable content templates is chosen first. Then, this content can be combined with purely random text to produce the synthetic content.

4. Apply styling templates: Depending on the domain of the synthetic dataset, a set of styling templates is first manually selected. Then, a style is randomly selected to format the appearance of the synthesized table.

5. Render the complete tables: The synthetic table is finally rendered by a web browser engine to generate the bounding boxes for each table cell. A batching technique is utilized to optimize the runtime overhead of the rendering process.

2. Prediction post-processing for PDF documents

Although TableFormer can predict the table structure and the bounding boxes for tables recognized inside PDF documents, this is not enough when a full reconstruction of the original table is required. This happens mainly due the following reasons:



Figure 7: Distribution of the tables across different dimensions per dataset. Simple vs complex tables per dataset and split, strict vs non strict html structures per dataset and table complexity, missing bboxes per dataset and table complexity.

- TableFormer output does not include the table cell content.
- There are occasional inaccuracies in the predictions of the bounding boxes.

However, it is possible to mitigate those limitations by combining the TableFormer predictions with the information already present inside a programmatic PDF document. More specifically, PDF documents can be seen as a sequence of PDF cells where each cell is described by its content and bounding box. If we are able to associate the PDF cells with the predicted table cells, we can directly link the PDF cell content to the table cell structure and use the PDF bounding boxes to correct misalignments in the predicted table cell bounding boxes.

Here is a step-by-step description of the prediction postprocessing:

1. Get the minimal grid dimensions - number of rows and columns for the predicted table structure. This represents the most granular grid for the underlying table structure.

2. Generate pair-wise matches between the bounding boxes of the PDF cells and the predicted cells. The Intersection Over Union (IOU) metric is used to evaluate the quality of the matches.

3. Use a carefully selected IOU threshold to designate the matches as "good" ones and "bad" ones.

3.a. If all IOU scores in a column are below the threshold, discard all predictions (structure and bounding boxes) for that column.

4. Find the best-fitting content alignment for the predicted cells with good IOU per each column. The alignment of the column can be identified by the following formula:

$$alignment = \arg\min_{c} \{D_{c}\}$$

$$D_{c} = max\{x_{c}\} - min\{x_{c}\}$$
(4)

where c is one of {left, centroid, right} and x_c is the x-coordinate for the corresponding point.

5. Use the alignment computed in step 4, to compute the median x-coordinate for all table columns and the me-

dian cell size for all table cells. The usage of median during the computations, helps to eliminate outliers caused by occasional column spans which are usually wider than the normal.

6. Snap all cells with bad IOU to their corresponding median *x*-coordinates and cell sizes.

7. Generate a new set of pair-wise matches between the corrected bounding boxes and PDF cells. This time use a modified version of the IOU metric, where the area of the intersection between the predicted and PDF cells is divided by the PDF cell area. In case there are multiple matches for the same PDF cell, the prediction with the higher score is preferred. This covers the cases where the PDF cells are smaller than the area of predicted or corrected prediction cells.

8. In some rare occasions, we have noticed that Table-Former can confuse a single column as two. When the postprocessing steps are applied, this results with two predicted columns pointing to the same PDF column. In such case we must de-duplicate the columns according to highest total column intersection score.

9. Pick up the remaining orphan cells. There could be cases, when after applying all the previous post-processing steps, some PDF cells could still remain without any match to predicted cells. However, it is still possible to deduce the correct matching for an orphan PDF cell by mapping its bounding box on the geometry of the grid. This mapping decides if the content of the orphan cell will be appended to an already matched table cell, or a new table cell should be created to match with the orphan.

9a. Compute the top and bottom boundary of the horizontal band for each grid row (min/max y coordinates per row).

9b. Intersect the orphan's bounding box with the row bands, and map the cell to the closest grid row.

9c. Compute the left and right boundary of the vertical band for each grid column (min/max x coordinates per column).

9d. Intersect the orphan's bounding box with the column bands, and map the cell to the closest grid column.

9e. If the table cell under the identified row and column is not empty, extend its content with the content of the or-

phan cell.

9f. Otherwise create a new structural cell and match it wit the orphan cell.

Aditional images with examples of TableFormer predictions and post-processing can be found below.

9.91 18.98 18.86 18.68 00dds ratio	▶67-9.14 ₿₽1-19.41 ₿⊅1-69.63 ₱₽₽-34.80 ₽₽ % confidence interval	(3.013 (40.001) (40.001) (40.001) (40.001) (40.001) (40.001)
18.98 18.86 18.68 odds ratio	821-19.41 351-69.63 288-34.80 295% confidence interval	4 €.001 4 €0.001 2 €0.001 2 €0.001
15.86 26.68 oding boxes Odds ratio	351-69.63 258-34.80 25% confidence interval	480.001 (220.001) (220.001)
vding boxes	25% confidence interval	@20.001
oding boxes Odds ratio	95% confidence interval	p value
16 9	67-914	64013
5.9	€ .67−9.14	0.013
NU.98	6921-19.41	au.001
13 .86	≴ ¢1−69.63	1\$0.001
12.68	18 38-34.80	19.001
oxes Odds ratio	95%) confidence interval	P value
9.91	b 67-9.14	2013
	80.98 183.86 183.68 0xes 0dds ratio	80.98 (62.1-19.41) 15.80 3.61-69.63 15.70 1287-34.80 xxxes 95.% 0dds.ratic 95.% interval interval (F7) (F7-91.4)

TableFormer predicted structure

Mevere bleeding

veurologic complications

٥ ۱۵. ۱۵ Variable	1 ព្រ.ឲ Odds ratio	2 (2, 0) 95% confidence interval	^{3 3,1} p value
4 (0, 1) Major vascular complications	5 (1. 1) 3.91	^{6 [2, 1]} 1.67–9.14	7 p. 0.013
8 (0, 2) Renal failure required CRRT	9 [1, 2] 10.98	10 [2, 2] 6.21–19.41	11 p.: < 0.001
12 (0, 3) Severe bleeding	13 (1. 3) 15.86	14 [2, 3] 3.61–69.63	15 p.: < 0.001
16 (0, 4) Neurologic	17 (1.4) 13.68	18 [2, 4] 5.38–34.80	19 p. 4 < 0.001

15.86

25.68

400.001

450.001

3.61-69.63

288-34.80

Figure 8: Example of a table with multi-line header.

PDF Cells	
Name	Sequences
KRAS =F	5-TGTGTGACATGTTCTAATATAGTCACATTT-3
KRAS	5'-ATCGTCAAGGCACTCTTGCCTAC-3
PNA clamp probe	5'-TACGCCACCAGCTCC-3
TableFormer predicted bounding boxes	
Name	² Sequences
KRAS-F	5'-TGTGTGACATGTTCTAATATAGTCACATTT-3
KRAS-R	5'-ATCGTCAAGGCACTCTTGCCTAC-3'
PNA clamp probe	5'-TACGCCACCAGCTCC-3
Post-processed bounding boxes	
Name	Sequences
KRAN	9-TGTGTGACAIGTTCTAAIAIAGTCACAITT-3
KKAN	9-AICGICAAGGCACICIIGCCIAC-3
PNA clamp probe	D-TACGCCACCAGCTCC-3

TableFormer predicted structure									
	0 (p, q		1.0						
Name		Sequences							
	2 (0, 1)		3 (1,						
KRAS-F		5'-TGTGTGACATGTTCTAATATAGTCACAT	TT-3'						
	4 (0, 2)		5 (1,						
KRAS-R		5'-ATCGTCAAGGCACTCTTGCCTAC-3'							
	6 (p, a)		7 (6,						
PNA clamp	probe	5'-TACGCCACCAGCTCC-3'							

Figure 9: Example of a table with big empty distance between cells.

ANOVA								
	Sum Sq	Df	⊮ Value	Fr (>F)				
P	3745.2	1	266.75	4964 10 18 334				
conc	2491.39	2	50.87	2976 20 20 232				
P 25 COnd	2648.33	2	61.48	1007 F1 1233				
Residuals	286.91	11	B ∎	Ba				
Residuals	286.91	И	} ≉8	₽ŧ				

0		2		
3	Sum Sq	5Df	F Value	Pr (>F)
₽	\$745.2	10	266.75	$4.64 imes 10^{-9}$
vonc	24191.39	2	50.87	2.76×10^{-6}
P1× conc	2648.33	20	61.48	$1.07 imes 10^{-6}$
Residuals	286.91	361	36	27

Post-processed bounding boxes

ANOVA								
	Sum Sq	Di	≥ Value	₽r (>F)				
E	3745.2	æ	266.75	4864 📷 1181 ²³ 44				
മോറ	1091.39	1	50.87	2976 20 PO 2 23				
🖄 😹 🙋 nc	2548.33	20	64.48	1907 (v 1961) 3 ⁽³⁴				
residuals	286.91	121	Be	Ba				

TableFormer predicted structure

	0* (0, 0)				1 (1, 0)	2* (4, 0
		ANOVA				
	3* [0, 1]	4 (1, 1]	5 [2, 1]	6 p. ŋ	7 [4, 1
		Sum Sq		Df	FValue	Pr (>F)
	8 (0, 2)	9 (1, 2)	10 (2, 2)	11 p. 2j	12 (4, 2
Р		5745.2	ŀ	1	266.75	4.64×10-
	13 (0, 3)	14 (1, 3)	15 (2, 3)	16 (3, 3)	17 (4, 5
conc		2191.39		2	50.87	2.76×10-
	18 (0, 4)	19 (1, 4]	20 [2, 4]	21 (p, 4)	22 [4, 4
P×conc		2648.33		2	61.48	1.07×10-
	23 [0, 5]	24 [1, 5)	25 [2, 5]	26 (p, s)	27 [4, 5
Residuals		236 91	- I-	11	_	_

Figure 10: Example of a complex table with empty cells.

PDF Cells			
	3 mM (ng/ml/islet)	16.7 (nM (ng/ml/islet)	Fold-increase (high GLC/ low GLC)
86 c <mark>ontrol</mark> 86 v id agliptin treated 86(A) ⁹ 200ntrol 16(A) ⁹ 16(dagliptin treated	0.047 ±0.07 0.018 ±0.06 0.034 ±0.004 0.080 ±0.07	3396 140327 4234 23826225 1406 180737 1481 379801	5053 9643927 96421
TableFormer predicted	bounding bo	xes	
	3 mM (ng/ml/islet)	36.7 mM (ng/ml/islet)	Fold-increase (high GLC/ lcw GLC)
186 control 186 vildagliptin treated 186KA ^y control ≇KKA ^y vildagliptin treated	0.247 ± 0.07 9748 ± 0.06 0x34 ± 0.04 0x30 ± 0.07	2.96±0.27 [4834±0.32* [2206±0.07*] [2k81±0.30↑]	6263 9343+ 9343+ 9343+ 62421
Post-processed bound	ng boxes		
	3 mM (ag/ml/islet)	16.7 mM (r.g/ml/islet)	Fold-increase (high GLC/ Low GLC)
ii6 cimtro] B€ viidagliptin treated KKA <mark>i≆be</mark> ntro]	0.047(±±0.007 0.048(±±0.007 0.054(±±0.007	1#261±40x27 4x641±41×622 1#061±41407097	526.3 5243357 5243359

TableFormer predicted structure

	0° (0, 6)	1" [1.0]	2* p. o	3 p.
				Fold-increase
	4° (0, 1)	5 (1.1)	6 (2. t	7 p.
	3ml	м	16.7mM	(high GLC/
	8° (0, 2)	9 (1.2)	10 (2, 2	11 p.
	(ng/	/ml/islet)	(ng/ml/islet)	low GLC)
	12 (0, 3)	13 (1. 3)	14 (2, 3	15 p.
B6control	0.43	7±0.07	2.96±0.27	6.63
	16 (0, 4)	17 (1.4	18 (2.4	19 p.
B6vildagliptin treated	0.48	8±0.06	4.34±0.32*	9.43+
	20 p. sj	21 (1.8)	22 p. s	23 (8.
KKAycontrol	0.34	4±0.04	1.06±0.07+	3.43+
	24 (0, 6)	25 [1.6]	26 (2, 6	27 p.
KKAyvildagliptin treated	0.30	0±0.07	1.81±0.30†	6.42†

Figure 11: Simple table with different style and empty cells.

PDF Cells						
Preatment		FCO ₂ (patm)		fotal alkalinity (#hol kg=1) ²²	Salinity (ppt)	Hemperature (°C)
Control	P	397 ±47. 5	896 2006	2145 z#7	35.6 27.07	28.6 20.05
Control	20	384 10.8	8718 30.006	2145 直纬7	55.6 ± 6 .07	28.4 90.04
Medium	•	614 3216.6	8.00 ± 8.009	2095 \$29.1	35.9 ± 8.07	28.7 ± 8.05
Medium	27	608 ±16.5	8.00 + 0.009	2095 ±99.1	35.9 37.07	28.6 10.05
High	•	876 3 74.6	7.86 30 .006	2079 \$2.3	36.0 30.03	28.7 30.03
(max)	~	1000 10 000 1	THE R. L. LEWIS CO.	10000 10 210	[222.0] x [22.20.0]	[474 3] - [746 4

TableFormer predicted bounding boxes

Treatment				Total alkalinity (µmol kg ⁻¹)		
Control		397±6.5	8/16±0.006	£2145 ± 4.7	135.6 ± 0.07	28.6 ± 0.05
Control	2	384±6.8	8:18±0.006	12145 ± 4.7	185.6±0.07	26.4±0.04
Medium	1	2614±16.6	8400±0.009	12095 ± 5.1	285.9 ± 0.07	28.7±0.05
Medium	2	#08±16.5	8:00 ± 0.009	2095 ± 5.1	335.9±0.07	28.6±0.05
thigh		876±14.6	3.86±0.006	2079 ± 5.3	486.0 ± 0.03	41 28.7 ± 0.03
(High)	-42	4861 ± 14.4	47.87 ± 0.006	42079 ± 5.3	4736.0 ± 0.03	428.7 ± 0.04

d bounding boxes

ireatment	lank number	FCO, (jiātm)	PH	fotal alkalinity	Salinity (ppt)	Hemperature (°C)
Control	6	397 308.5	896 30.006	2145 世纪7	35.6 30.07	28.6 30.05
26ntro	g.	384 100.8	8718 00.006	2145 3477	35.6 0.07	28.4 90.04
Medium	C*	914 3496.6	8:00 3 0 .009	2095 345.1	35.9 B.07	28.7 38.05
Medium	B	508 3976.5	8:00 10:009	2095 38:1	35.9 30.07	28.6 00.05
Pfigh	0*	876 JAP4.6	7:86 00.006	2079 38:3	58.0 30.03	28.7 30.03
Mah	201	86R (3044.4	7.87 60.006	2079 389	3610 100703	287 10004

TableForm	ableFormer predicted structure												
	0 (0, 0)	1 (1.0		2 (2, 4)	3	(A, 0)		4 (4, 0		5 (5, 0)		6 (9	
Treatment		Tank number	PCO\$_{2}\$ (µatm)		pН		Total alkalin (µmol kg-1)	ity	Salinity (ppt))	Temperature (*C)	Э	
	7 (0, 1)	8 (1, 1)		9 (2, 1	10	p. 0		11 (6, 1		12 (5, 1)		13 (5	
Control		1	397 ± 6.5		8.16 ± 0.006		2145 ± 4.7		35.6 ± 0.07		28.6 ± 0.05		
	14 (0.2)	15 (1. 2		16 (2, 2	17	(A. 2)		18 (4, 2		19 (5.2)		50 la	
Control		2	384 ± 6.8		8.18 ± 0.006		2145 ± 4.7		35.6 ± 0.07		28.4 ± 0.04		
	21 jp. s	22 (1.3)		23 (2, 3)	24	(D. 8)		25 (4. S		26 (5.3)		27 (5	
Medium		1	614 ± 16.6		8.00 ± 0.009		2095 ± 5.1		35.9 ± 0.07		28.7 ± 0.05		
	28 (0.4)	29 (1.4		30 (2.4)	31	(p. 4)		32 (4, 4		33 (5.4)		34 (5	
Medium		2	608 ± 16.5		8.00 ± 0.009		2095 ± 5.1		35.9 ± 0.07		28.6 ± 0.05		
	35 (p, s)	36 (1, 5)		37 (2, 5)	38	(3, 6)		39 (4, 5		40 (5, 5)		41 (5	
High		1	876 ± 14.6		7.86 ± 0.006		2079 ± 5.3		36.0 ± 0.03		28.7 ± 0.03		
	42 (0.6)	43 (1, 4		44 (2.4)	45	p. 6		46 (4, 6		47 (5.6)		48 (9	
High		2	861 ± 14.4		7.87 ± 0.006		2079 ± 5.3		36.0 ± 0.03		28.7 ± 0.04		

Figure 13: Table predictions example on colorful table.

ļ	PDF Cells			
	Variable	Sensitivity (%)	Specificity (%)	Cut off
	Fotal Bilirubin	60	95	₽,3 mg/dl
	Direct Bilirubin	60	95	0,85
	🚱 Reactive Protein	47	85	96
	TableFormer predicted bounding box	es		
	d/aviable	Fancitivity (9/)	(Encolficity (9/))	Cut off

-v ur iubie	Sensitivity (70)	opecificity (70)	C ut on
Total Bilirubin	*60	95	1,3 mg/dl
Direct Bilirubin	*60	95	10,85
¹⁴ C- Reactive Protein	147	85	198

essed bounding boxes

V ariable	Sensitivity (%)	Specificity (%)	©ut off
Fotal Bilirubin	60	92	F ,3 mg/dl
Direct Bilirubin	60	96	8 85
Reactive Protein	1 7	85	98

TableFormer predicted structure

0 (8.0)	1 (1.0)	2 (2.0)	3 (1.9
Variable	Sensitivity (%)	Specificity (%)	Cut off
4 (0, 1)	5 (1, 1)	6 (2, 1)	7 (0, 1)
Total Bilirubin	60	95	1,3 mg/dl
8 (0. 2)	9 (1, 2)	10 (2, 2)	11 (0.2)
Direct Bilirubin	60	95	0,85
12 (0.3)	13 (1, 3)	14 (2, 3)	15 (1.3)
C- Reactive Protein	47	85	98

Figure 12: Simple table predictions and post processing.

ľ	PDF Cells					
	6 ortical Layer	Grade 4	Grade 3	Grade 2	Grade 1	
	Molecular	Finiformly thick and rellular	Variable thinning, normal cellularity	Variable thinning and reduced cellularity	teniformly thin	
	Burkinje	well populated with histologically intact pyramidal neurons	[molated neuronal loss or] mosinophilic degeneration] (mecrosis)	bioderate gaps and anattered loss of neurons	barge gaps and conspicuously increased peuronal necrosis	
	Granule	Gensely cellular	densely cellular	aregular thinning with modest reductions in cell density	aregular thinning and conspicuous reductions in coll density	

r predicted bounding boxe

•Cortical Layer	Grade 4	*Grade 3	Grade 2	Grade 1
Molecular	© Uniformly thick and cellular	Variable thinning, normal cellularity	*Variable thinning and reduced cellularity	[Uniformly thin]
Purkinje	Well populated with Histologically intact pyramidal neurons	Lisolated neuronal loss or eosinophilic degeneration (necrosis)	Moderate gaps and scattered loss of neurons	*Large gaps and conspicuously increased neuronal necrosis
Granule	Uniformly thick and densely cellular	Prregular thinning but densely cellular	^a Irregular thinning with modest reductions in cell density	¹³ Irregular thinning and conspicuous reductions in cell density

rocessed	bounding boxes	
ical Layer	Grade 4	67

1	Post-processed bounding boxes												
	ortical Layer	S rade 4	brade 3	Grade 2	Grade 1								
	Molecular	eniformly thick and selfular	wanable thinning, normal selfularity	manable thinning and reduced cellularity	Sanformly thin								
	farkinje	mell populated with histologically intact nyramidal neurons	necrosis)	nioderate gaps and mattered loss of neurons	parge gaps and conspicuously increased pauronal necrosis								
	Scanule	conformly thick and densely cellular	aregular thinning but lansely cellular	nodest reductions in cell density	megular thinning and panspicuous reductions in pell density!								

TableFormer predicted structure

0 (0.0)	1 (1.0	2 p. 0	3 [5.0]	4 (4, 0)
Cortical Layer	Grade 4	Grade 3	Grade 2	Grade 1
5 (0, 1)	6 (1, 1	7 (2, 1	8 (3, 1)	9 (6.1)
Molecular	Uniformly thick and cellular	Variable thinning, normal	Variable thinning and	Uniformly thin
		cellularity	reduced cellularity	,
10 p. g	11 (1.2)	12 p. z	13 p. g	14 p. g
Purkinje	Well populated with	Isolated neuronal loss or	Moderate gaps and	Large gaps and
	histologically intact	eosinophilic degeneration	scattered loss of neurons	conspicuously increased
	pyramidal neurons	(necrosis)		neuronal necrosis
15 p. a	16 p. a	17 µ.s	18 p. a	19 p. a
Granule	Uniformly thick and densely	Irregular thinning but densely	Irregular thinning with	Irregular thinning and
	cellular	cellular	modest reductions in cell	conspicuous reductions in
			density	cell density

Figure 14: Example with multi-line text.

PDF Calle															Parameter	Value		
	1	Size Grade Involved			BR	MER-2	I-2 Ki-67	6VI	MVD	Loco-regional	Systemic				Gain	2851 V/V (69.09	9 dB)	
				lymph Rode	status	status	status			treatment	treatment			Low	Cut-Off Frequency	285 Hz		
Ange (3400/>50 year	5]	(Hewalue) 36058	B236	(#+value) (8:495	(aqualue) (82:97	3:054	(#walue) (3:641	(Movalue) (34913	(Movalue)	(Mivalue) (98261	(#svalue) <0.001			High	Cut-Off Frequency	6580 Hz		
See (#10/21-50/>50) mm)		0:033	<0.001 9£58	(8 9 679) (9 4 753)	6626 •0.001	@£041] <0.001]	<0.001	0.672	*0.001	0x438 0x627			Inp	out-Referred Noise	2.1 #V (rms)		
wolved lymph					01157	9.623	0035	40.001	9:607	0:001	81013				emrr	1210 dB (a) 1 KHz		
Boogesterone	ptor					6.002	0£40	3692	0.097	31256	84804			Numb	er of Analog Channels	24		
100R-2 status							0.620	1.001 9	0.627	0.063	8,603			Po	wer Consumption	14 mA @ 3.0 V (3	mW)	
1067 proliferative a	ctivity 67)							0.049	0.041	0.698	8.089				Precision	Selectable, 12 or	8 bits	
(Absent/Present) (Absent/Present) (Absention (Absent) (Absention (Absent) (Absention (Absent) (Absention (Absent) (Absent/Present)									1991 5	(3:333) (3:322)	133855 133802 133867			TableFormer predicted bounding boxes				
stectomy + radiot	therapy)														Parameter	Value		
TableFormer prec	dicted bou	inding bo	oxes												Gain	2851 V/V (69.09	9 dB)	
0	1	Size	² Grade	Ivolved lymph node	PR4 5 status	HER-2 status	⁶ Ki-67 status		MVD	Loco-regional treatment	多ystemic treatment			Low	Cut-Off Frequency	285 Hz	,	
11		(P-value)	(P-value) (P-value)	(P-value)	(P-value)	P-value)	#(P-value)	P-value)	(Pevalue)	(#-value)			Hiet	Cut-Off Frequency	6580 Hz		
sage (≤30/>30 years	5/) mm) 34	10000	\$1033	×0.001	0.200	10 .034	0 041	40.001	94258 94568	40.001	0.001			Inr	out-Referred Noise	$2.1 \mu\text{V}$ (rms)	
Prade (1/11/11) Provided lymph	45	3	46	49.258	0. P663] 0. 1597	*0.001	40.001	\$0.001 \$0.001	0x072 0x607	B#08 6:001	3x627 3x613					11760 dB @ 1 K	Hz	
node (0/1-3/>3) Progesterone -rece	ptor 67	[68	69	70	0.002	08340	₹ 7 .392	8.997	8256	7804			Nuthb	er of Analog Champels	2		
status (PR-/PR+)	78	[79	80	81 82		6 .020	8 .119	6 .627	0.063	0803			Pe	wer Consumption	$1 \frac{1}{100} $	mW)	
(HER-2 -/HER-2+)		50		91	- (983		94	99 049	09041	07598	01289			1	Precision	Selectable 12 or	$\frac{1}{1} \frac{1}{1} \frac{1}$	
(Low Ki-67/High Ki-67) 140 MI (Absent/Present) MVD (tertile[17, 2, 3] Loco-reg[178] treatment		112	01] 10 113	12 105		Like Like	305 10 116 3	17	(3.0915) [118	[3:0931] [3:0628]	0.0855 0.002 0.0857			Post-processed bounding boxes		Beleetable, 12 of	0 010	
(Edmpectomy + radiotherapy) mastectomy + radiotherapy)															Parameter	Value		
Post-processed t	bounding	boxes													Gain	2851 V/V (69.09	9 dB)	
		size	Grade	nvolved	R	HER-2	di-67	Ξ¥.	MIVD	aoco-regional	systemic	i		LOW	Cut-Off Frequency	285 Hz		
		(Hwalue)	Me value	node	aluel	Biwalue	litimatue	Heralue	limatue	Mavalue	(Amalue)			High	Cut-Off Frequency	6580 Hz		
%g e (36 ≉0/>50 year	rs.	96 058	84230	3 9495	86297	99054	90541	9.012	\$92.3 2	98261	e0.001			in the second se	out-Referred Noise	2.1 @V (rms)		
made ((/11/11))	J mm,		8033	80.00 I 82258	99675 99753	9620 40.001	#0.001	10.001	\$6008 \$6072	BRIDE	8627				MRR	$\frac{2.1}{4}0 \text{ dB} (a) 1 \text{ KHz}$		
Nolved lymph Cole (0/1-3/>3)					8 9 57	0.623	0035	40.001	\$1607	0.001	91013			Numb	er of Analog Channels	(
Hogesterone (1990) Hatus (PR-/PR+)	ptor					8,002	96 40	9692	38 /97	96256	9604			Po	wer Consumption		mW)	
100R-2 status 100R-2 -/HER-2+)							\$62 0	e rank	Add27	0.0853	\$0603			10	Precision	Selectable 12 or	8 hits	
967 proliferative a 1999 Ki-67/High Ki-	-67)							\$36945	54641	00098	54465				riccision	Sciectable, 12 of	0 0115	
Absent/Present MoD (tertiles 1, 2, 3 Sauc-regional treatm Lampectomy + radi	t Inent Iotherapy/								1.0012	56031 55622	30055 30002 30057				TableFormer predicted struc	cture		
rzzstectomy + radiol	therapy)														⁰ ₪.0 Parameter	l 1 [1. 0] Value		
bleFormer predicted s	structure ize	Grade	2 p. q	volvedivmohno	n.e de PRstatus	4xe	5m. R-2status	Ki-67status	*ו	77.4 MVD	**** L 000+	\$ K.G	10px e Systemictreatment		2 (0, 1	3 (1, 1)		
11° p. 1	12 2 amiuo)	(P. solution	13 p. 1	torrouginprinc (4	p. tj	15 K 1	16 p. 1	(Puratura)	(Ruoluo	Mp. 1 (Ramium)	region 19 p. 1	altreatment	21 (Pauraluo)		Gain 4 (p. 2	2851 V/V (69.09 dB)		
22 p. 2 ge (<50/>50 years) 0.	.058	0.236	24 p. a 0.	495	0.297	24 jk 2 0.05	27 (s. 2	0.641	0.913	24 y. a 0.238	30 p. 3 0.261	31 (6.2)	32 (N. 8 <0.001		Low Cut-Off Frequency	285 Hz		
ize (≤20/21-50/>50 m)		0.033	<	0.001	0.879	0.8	26	0.041	<0.001	0.568	<0.001	1	0.438		6 (0.3) High Cut-Off Frequency] 7* [1, 3]		
rade (VII/III)		0.8	астр. с 0. 87° µ. 6	258	0.753	<0.0	001	<0.001	<0.001	0.072	0.108	64 ju s	0.627	8° (0, 4)		9 (1, 4)		
I/1-3/>3)	67*	p. 4	64° p. e	<i>w</i>	0.157 p.e	0.33 ^{30' p. e}	71 p.4	0.035	<0.001	0.607	0.001	75 p. e	0.013 76 pa. e	6580 Hz ¹⁶ /10 (0.5) 20		6580 Hz		
rogesterone- ceptorstatus (PR- PR+)						0.00	02	0.340	0.392	0.997	0.256		0.804	Input-Referred Noise 2.1µV (rms)				
ER-2 status(HER-2 - (ER-2+)		0.8	70° µ. 1)	807	p. 0	611 (K. 1)	62° p. r	0.020	0.119	0.627	0.363	66 p. 1)	0.603	12 (0, 6) 13 (1, 6) 12 (0, 6) 13 (1, 6) 13 (1, 6) 14 MPz				
i-67 proliferative ctivity(Low Ki-67/High			9.4	91	1	~ 55	ALC D.		0.049	0.041	0.598	W K G	0.289		14 [0, 7	15 [1, 7]		
/I (Absent/Present)	100"	0.0	101' µ. 9	102	p. 9	100° je je	904° (s. s	10	r p. o	0.215	0.031	106 p. ej	0.255		Number of Analog Channels	2 1 17 (1, 8)		
VD (tertiles 1, 2, 3)	122*(7.79	123° (J. 10	124.	5.99	125° (4.10	128" (5. 11	129	- M. 10	128° p. 19	0.628	180* (4.10)	0.402		Power Consumption	1mA @ 3.0 V (3 mW)		
satment(Lumpectomy radiotherapy/													0.001	Precision Selectable, 12		Selectable, 12 or 8 bits		

PDF Cells

Figure 15: Example with triangular table.

Figure 16: Example of how post-processing helps to restore mis-aligned bounding boxes prediction artifact.

PDF Cells				TableFormer predicted bounding boxes				Post-processed bounding boxes				TableFormer predicted structure			
kem		Farmers	General community	ltem		Farmers	² General community	Item		tarmers	² General community	ltem	0 p. 3* p.	Farmers	Generalcommunity
Socio-e	conomic			Sociore	conomic	4	5	Socio-e	economic			6 p.	7 p. 18-29 ure	8 20	9 p.a 12.55
Age (%)	18-29 yrs	9.20	2.55	Age (%)	18-29 yrs	8.20	2.55	age (%)	#8-29 yrs	1 .20	2.55	10* 10	10-25 yrs 11 p.	120	13 p. s
	30-49 yrs	47.80	38.41	10	49 yrs	10/.80	38341		第 0-49 yrs	10 080	55.41	14° p.	30-49 yrs 15" p.	47.80	38.41
	98+ yrs	44.00	49.05	14	309+ yrs	44.00	49.05		58 + yrs	44.00	49.05	58.00	s 19 r.	44.00	49.05
5MEX (96)	Male	12.80	49.96	S& × (96)	Male	20 72.80	21 58:951	10 × (96)	Phale	12.8 0	48.96	Sex (%)	50+ yrs	72.80	
	Fiemale	20.20	50.04		Semale,	27.20	560.04		se male	20.20	20.04	27.0	Male	4 A D.	49.96
acation (%)	Major metropolitari	Q 4	28 .35	Ebcation (%)	Major metropolitan		58:35	20 cation (%)	Major metropolitar	32 4	58.35	Location (%)	FemaleMajor metropolitar	28 9.	29 p. 1 50.0458.35
	Remainder of State	220.00	40.65		Remainder of State	100.00	和 .65		Remainder of State	19 0.00	41.65	30' p.	Bemainder of State	10100.00	33 p.e
Relationship Sectus (%)	Married	#5.80	36.19	Relationship	Married	\$6.80	98.19	selationship stetus (%)	Married	A6.80	56 .15	34 p. Relationshipstatu: (%)	Married S5 p.	75.80 36 p.	56.19
	single	34.20	46.81		Sangle	24.20	43.81		angle	#1 .20	46.8	30° R.)	29 p	40 (2.1	42.01
Work status (%)	Rall-time	39.60	40.23	Work status (%)	Full-time	80.60	42.23	Mork status (%)	mall-time	\$\$.60	Aut.2.5	42 p. 1	alligie 43 p.	24.20 rt 44 p.1	45.01 45 p. m
	Rart-time	19.40	99.12	40	Part-time	16.40	19.12		Part-time	19.40	¥9.12	Work status (%)	Full-time 47 p.	80.60	42.23 49 p. sp
	Not in paid work	-as	38.65	50	Not in paid work	54	38.65		Not in paid work	Ba	36 .65	50° n 1	Part-time	19.40	19.12
evel of	Begree	\$4.20	\$4.75	Sevel of education (96)	Segree	14.20	31 .75	movel of Mucation (%)	Regree	14.20	54.75	54 k 1	Not in paid work	- 56 p. 14 20	38.65 57 p. w
	Mploma 1	38.80	55.83	58	Diploma	50 38.80	99 .83		Moloma	\$8 ,80	55.85	(%)	Logico	14.20	51.75
	Wear 11	19.60	\$6.38	62	Par 11	(189 .60)	1633		Pear 11		96.38	50° (t. 1	Diploma	38.80	33.83
	¥ear 10	W.20	\$0.31	66	Mar 10	FF.20	1031		Mear 10	W .20	14.31	62° ju s	4 63 p	20.60	66 p. w
	Year 9 or below	920	673		Vear 9 or pelow	73.20	773		Wear 9 or below	\$12 20	87/2	66' p. (Vaar 10	17.20	n 600 p. 11 11 31
eusehold income	44\$30,000	\$7.40	# 4.03	Plousehold income	< 5 \$30,000	79 .40	299.03	eusehold income	₩ \$30,000	W .40	M4.03	70° µ. s	Year 9 or below	9.20	73 p. m 6.73
	\$ \$ 0,000- \$1 9,999	59.20	48.81	70	73 0,000-\$79,999	50.20	42.81		\$\$0,000<u>-</u>\$ \$9,999	99. 20	42661	Household	< \$30,000	17.40	21.03
	980,000+	39.40	36.16		\$80,000+	32.40	36.16		980,000+	32 .40	36.16	78° p. 2	79 p.:	80 p. z	119 p. 20
Satis	faction			01 Satis	faction			Satis	faction				\$30,000-\$79,999	50.20	42.81
Gwerall (M, SD)		34 .55 ³⁴ (£7.83)	17.25 (#6.64)	Øverall (M, SD)	84	7 4% 55° (17.88)	77.25°(16.64)	øverall (M, SD)		34.55 (67.83)	A7.25 (#0.64)	Satisfaction	a 84° a :	a 65 a.a	8 5.22
	No.		SET 2 (5577)							No. of Marco and		Overall (M, SD)		74.55b(17.83)	77.25(16.64)
(Mornain (M, SD)	mmunity	\$2.5/]*[\$80.04)	00.21 (BC32)	Bomain (M, SD)	Connectedness	9.87" (18.64)	69. 5% (19.92)	pomain (M, SD)	sonnectedness	#8.07 Piec.04	00 .57 (10.92)	Domain (M, SD)	Communityconnectednes	s 73.87a(18.64)	69.57(19.92) 94 p. m
	Relationships	93.17 *(*7.63)	39.49 (32.79)	31 35	Relationships	83.17 ^a (17.63)	7949 (22.79)		Relationships	83.17 47.63	19.49 (32.79	95' 8.2	Relationships	83.17a(17.63)	79.49(22.79)
	Safety	33.61 ³ (355.54)	18294 (18264)		Safety	83.61 ^a (16.54)	78994 (17.84)		satety	33 .61 4(866 .54)	ANK 94 ([MAK4]		Safety	83.61a(16.54)	78.94(17.84)
	Stendard of living	14\$46 1(\$07.91)	77466 ³ (20.65)	99	Stendard of living	74346 ^b (16.91)	77:66° (16: 65)		stendard of living	34346 ³ [\$60.91]	21866" [20.65]	W K3	Standard of living	74.46b(16.91)66.8	977.66a(16.65)
	Return security	368 39 ³ (\$18.34)	X4539 (20612)	103	Future security	86.89° (21.34)	79939 (20.12)		terure security	38859 [[ba.34]	X4439 (200412)	103° ji 2	Future security	b(21.34)76.80	71.39(20.12)
	Hugalth	16680 167	34133 (19822)	1HI	kiealth	10 80° (17.67)	74.33 (19.22)		MERGITI	10830 1 00 .67	74438 (199822)	107° je 2	106 p.;	9(17 67)	74 33/19 22)
	Ashieving in life	12136 (18834)	74:09 138:54		Achieving in life	72.36 (18.34)	74.09 ^a (18.54)		Ashieving in life	12936 (198934)	74109 148.54	111° p. z	112 p.:	113 p. 2	114 p. 24
	Religion/spirituality	(15034 1893.24)	68823 ^{[1} [96.21]		Religion/spirituality	65.34 ^b (26.24)	68.23 ^b (26.21)		Religion/spirituality	356 34 1 (00 .24)	68234196.21	115° g. s	Achieving in life	12.36(18.34)	74.09a(18.54) 118 p. xi
													Religion/spirituality	65.34b(26.24)	68.23b(26.21)

Figure 17: Example of long table. End-to-end example from initial PDF cells to prediction of bounding boxes, post processing and prediction of structure.