End-to-End Compound Table Understanding with Multi-Modal Modeling

Zaisheng Li*
Hikvision Research Institute
Hangzhou, China
lizsh1@shanghaitech.edu.cn

Yi Li*
ShanghaiTech University
Shanghai, China
liyi3@shanghaitech.edu.cn

Qiao Liang*†
Zhejiang University & Hikvision Research Institute
Hangzhou, China
qiaoliang6@hikvision.com

Pengfei Li
Hikvision Research Institute
Hangzhou, China
lipengfei27@hikvision.com

Zhanzhan Cheng
Hikvision Research Institute
Hangzhou, China
chengzhanzhan@hikvision.com

Yi Niu
Hikvision Research Institute
Hangzhou, China
niuyi@hikvision.com

Shiliang Pu
Hikvision Research Institute
Hangzhou, China
pushiliang.hri@hikvision.com

Xi Li†
Zhejiang University
Hangzhou, China
xilizju@zju.edu.cn

ABSTRACT

Table is a widely used data form in webpages, spreadsheets, or PDFs to organize and present structural data. Although studies on table structure recognition have been successfully used to convert image-based tables into digital structural formats, solving many real problems still relies on further understanding of the table, such as cell relationship extraction. The current datasets related to table understanding are all based on the digit format. To boost research development, we release a new benchmark named ComFinTab with rich annotations that support both table recognition and understanding tasks. Unlike previous datasets containing the basic tables, ComFinTab contains a large ratio of compound tables, which is much more challenging and requires methods using multiple information sources. Based on the dataset, we also propose a uniform, concise task form with the evaluation metric to better evaluate the model's performance on the table understanding task in compound tables. Finally, a framework named CTUNet is proposed to integrate the compromised visual, semantic, and position features with a graph attention network, which can solve the table recognition task and the challenging table understanding task as a whole. Experimental results compared with some previous advanced table understanding methods demonstrate the effectiveness of our proposed model. Code and dataset are available at https://github.com/hikopensource/DAVAR-Lab-OCR.

CCS CONCEPTS

• **Computing methodologies → Object recognition.**

KEYWORDS

Dataset, Table Understanding, Multi-Modal Learning

ACM Reference Format:
Zaisheng Li, Yi Li, Qiao Liang, Pengfei Li, Zhanzhan Cheng, Yi Niu, Shiliang Pu, and Xi Li. 2022. End-to-End Compound Table Understanding with Multi-Modal Modeling. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), October 10–14, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3503161.3547885

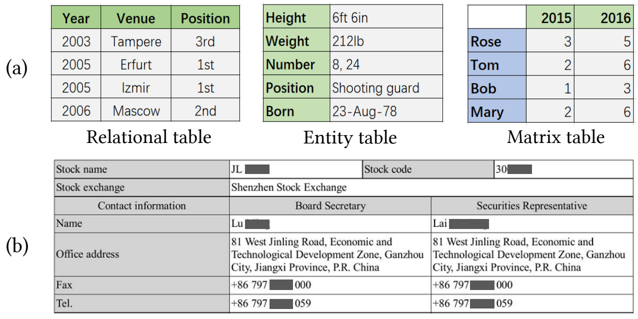
1 INTRODUCTION

Table as a key structure to organize and present data is widely used in webpages, spreadsheets, PDFs, *etc.* Studies on table detection [26, 32, 33, 41, 44] and table recognition [23, 41, 44–46, 66, 74] have attracted great attention and been successfully used to convert image-based tables into the digital structural formats. However, the conversion results usually only contain the structure-level information. Solving many real problems still relies on a further understanding of the table content, such as extracting the entities with the identified attributes in the table.

The current researches on table understanding are mainly under relatively simple settings, which can reflect in two folds. First, studies are mainly conducted based on the digital format of tables like Spreadsheet or XML [12, 17, 18, 29, 43, 69, 73, 73]. These digital formats whether only contain the limited textual information [5, 6] or rely on the handcraft

*Zaisheng, Yi and Liang contributed equally to this research. Yi did this work during an internship in Hikvision Research Institute.
†Corresponding authors.

MM '22, October 10–14, 2022, Lisboa, Portugal
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-9203-7/22/10. . . $15.00
https://doi.org/10.1145/3503161.3547885



(a)

Year	Venue	Position
2003	Tampere	3rd
2005	Erfurt	1st
2005	Izmir	1st
2006	Mascow	2nd

Relational table

Height	6ft 6in
Weight	212lb
Number	8, 24
Position	Shooting guard
Born	23-Aug-78

Entity table

	2015	2016
Rose	3	5
Tom	2	6
Bob	1	3
Mary	2	6

Matrix table

(b)

Stock name	JL [redacted]	Stock code	30[redacted]
Stock exchange	Shenzhen Stock Exchange		
Contact information	Board Secretary	Securities Representative	
Name	Lu [redacted]	Lai [redacted]	
Office address	81 West Jinling Road, Economic and Technological Development Zone, Ganzhou City, Jiangxi Province, P.R. China	81 West Jinling Road, Economic and Technological Development Zone, Ganzhou City, Jiangxi Province, P.R. China	
Fax	+86 797 [redacted] 000	+86 797[redacted]000	
Tel.	+86 797 [redacted] 059	+86 797[redacted]059	

Compound table

Figure 1: The examples of (a) three types of basic tables, which are more easily to extract the information based on the structural information, and (b) the more challenging compound table, in which the cells belong to the same row/column may have no semantic relationship.

features [9, 14, 30, 62] like color, font, *etc.* However, these features cannot be directly obtained for an image-based table unless designing extra information extraction tasks, making the table understanding process in images complicated and customized. Second, most current works only study the basic and regular table forms. In [62], authors categorize tables into three types: relational, entity and matrix tables, as shown in Figure 1(a). The main basis of classification is whether the table contains the head keys in column/row. However, tables in the real world can have more varied and free structures. For example, Figure 1(b) demonstrates a table from a financial report. This table is actually compounded by several small basic tables with different types, making the relationships among cells complex and difficult to extract by simple heuristic rules.

The current public datasets of tables can be roughly divided into the image-based [15, 16, 19, 32, 34, 38, 74] and the digital-format-based [12, 17, 18, 29, 43, 69, 73, 73]. Although some of them contain the challenge table with complicated structures when cells cross span multiple rows/columns, they still cannot fully cover scenarios in real life that require tabular understanding. On the one hand, these image-based datasets mostly focus on visual perception tasks like structure recovery. On the other hand, in both types of these datasets, the vast majority are basic tables.

To boost the development of research on table understanding and help the technique support a more comprehensive range of tables in the actual product, we collect a new image-based benchmark named ComFinTab that supports both table recognition and understanding tasks. The dataset contains 10k images collected from the public financial statements of the listed companies in two languages (4k English and 6k Chinese). We carefully select the represented tables to make the dataset involve more than 70% complex compound tables. We provide complete annotations, including the textual location, textual content, cell location, cell type, and cell relation. All annotations are automatically extracted by the existing tools or processing scripts and then manually rectified by people.

Table understanding is somehow a broad concept that might include different task forms such as table type classification [9, 14, 30, 62], cell type classification [17, 20, 58, 62], column type identification [3, 21], entity linking [27], and Table QA [25, 43, 68]. However, these task forms are not unified under different datasets or settings. For example, although both are cell classification tasks, [17] and [12] have different cell category definitions. We also find that some tasks can be derived from each other to some extent. For example, we can easily infer the table type and entity's links by obtaining the cells types in the basic tables. Therefore, we attempt to establish a concise and unified task form and evaluation metric for the proposed dataset. Specifically, we define the basic table item as a tree, where the root node is a "data" cell, and the left and right sub-tree store the "left header" cell information and "top header" cell information, respectively. Each basic table item represents a piece of information, which together constitute all the information conveyed by the table.

To solve the challenge table understanding problem in the compound tables, in this paper, we propose a framework named CTUNet (Compound Table Understanding Network) that utilizes different modality features, including visual, textual, and position information. The idea is mainly based on the findings that people can utilize multiple dimensions of information to understand complex tables. In addition to the basic alignment features, many other visual clues can also help people quickly obtain the information, such as the cells' color shown in Figure 1(b). Nevertheless, even if there is no separable color, we can still understand the table to a large extent since the text can also deliver important messages. Therefore, in CTUNet, we establish features that fuse multiple modalities for each cell and then make full use of the structural characteristic of the table to fuse and transfer these features by a Graph Attention Networks [60]. The final items can be directly inferred by the node and edges categories of the output graph. Experiments compared with the previous modeling methods demonstrate the effectiveness and robustness of our proposed framework.

The major contributions of this paper are as follows: (1) We establish and release a new image-based dataset with rich annotations for the table understanding tasks, including many challenging compound tables. (2) We propose a concise and uniform task form with the evaluation metric for the table understanding task, which can be applied to almost all previous datasets. (3) We propose a novel table understanding framework that utilizes visual, textual, and position features. The experimental results demonstrate its effectiveness and robustness.

2 RELATED WORK

2.1 Table Recognition

Image-based table recognition task aims to obtain the structural information with the content in the table. Due to the challenge of the table with varied structures, most works related to table recognition focused on the structure recognition task and directly adopted the existing OCR engines [13, 53].

Current table structure recognition methods can roughly be divided into three types. The first group of methods [49, 51, 52] starts by detecting the row and columns of a table and then merges these two parts to obtain the cells. To handle the cells crossing multiple rows/columns, [59, 71] predicts another indicator to merge the separated cells. The second group of methods [8, 28, 34, 37, 38, 45–47, 66] first detects the cells or text regions and then predicts the relations among these items. According to the underlying alignment information in the table, [38, 46, 47] aim to obtain more accurate aligned cells which can be effectively used to infer the final structure. [34, 37, 45, 66] treat these cells as nodes in a graph and train another Graph Neural Network (GNN) to predict the relations. The last group of methods [67, 74] directly obtain table recognition results from the original image based on an encoder-decoder architecture. [74] first uses the Image-to-Sequence architecture to predict a long sequence representing table structure and content together. To address the problem of sequences being too long, [67] predicts sequences only representing the table structure and provides position predictions for each cell identifier in the sequence. A separate OCR model obtains the text contents.

Table structure recognition technology has made significant progress in recent years, partly thanks to the public of some valuable datasets [8, 15, 16, 19, 32, 34, 38, 50, 54, 72, 74]. For example, [74] first releases a large-scale scientific table benchmark that contains a large part of gridless challenge tables, which effectively promotes the development of data-driven methods. [38] first studies the tables in the natural scene, requiring a higher model detection accuracy. All the above works are primarily based on the basic tables, and extracting the information can sometimes support by simple rules. However, in the compound tables, cells with the same column/rows might match with keys, which is a challenging scenario.

2.2 Table Understanding

Current research on table understanding can be roughly divided into two categories according to the purpose of the task.

The first type of method aims to parse the natural language in the tabular data, where the table appears as a carrier of texts. There are two typical types of task forms: 1) *table question answering(Table QA)*, which is a task that requires models to understand tables and natural language questions jointly and enable robust reasoning over tables. These works are mainly focused on the table with relatively simple structures such as the relational table in database [61, 69, 73] or relational web tables [43, 57]. 2) *Table-to-text*, whose target is to generate textual descriptions from structured tables [1, 2, 4, 31, 35, 40, 42, 55, 63]. For example, NumericNLG [55] proposes to generate reasoning-aware paragraph-level descriptions for tables in scientific papers. HiTab [7] firstly introduces hierarchical tables as the generation context, posing new challenges compared with previous datasets.

The other category of the method is usually built based on the structural information of the table, which aims further to determine the function of cells and their logical relationships. These works mainly contain the task forms like *table type classification* to categorize tables into different structural types [11, 18, 39, 62], cell type classification to identify cell types in the table [12, 17, 29], and *cell linking* to map the keys in a table with the data items [27]. Although the above tasks have different task targets, they can be converted to each other in the case of all basic tables.

3 TABLE UNDERSTANDING TASK ON COMFINTAB

3.1 Dataset Collection

The current public table datasets are primarily focused on the *basic tables*, where all the keys are only displayed on the top or left of the table, *i.e.,* relational table, entity table, and matrix table [62]. To investigate the table understanding task in the more general and complicated situations, we collected a new dataset named ComFinTab. The dataset contains 10,000 images from the annual reports of the Chinese listed companies in both the Shanghai and Shenzhen Stock Exchanges. Some companies have released annual reports in both Chinese and English versions so that the dataset can be separated by language as 6,000 Chinese and 4,000 English. Different from previous tables, we pay more attention to the *compound tables* images. A compound table is a table that is integrated by more than two basic tables A comparison of ComFinTab with previous representative benchmarks is shown in Table 1.

ComFinTab provides annotations that support both table recognition (cells local, cells column/row indexes, textual position, and textual content) and table understanding tasks (cells type, cells linking). The annotations are firstly generated by the tools and manual scripts and then carefully adjusted by people. Specifically, we can summarize the process in five steps: 1) We first locate the tables and crop them from the PDF files using a bordered tables detector, and cell locations can also be obtained by crossing the lines in tables. Some error predictions will be filtered out manually. 2) Since all of the tables are horizontally displayed, we adopt the cell matching strategy in [46] to generate the column/row indexes. 3) The PDF tools (like PDFPlumber) can easily extract the textual locations and contents. 4) For the cells type, we first select the tables that contain cells in different colors and automatically label them. Then, we use these samples to train a simple text classifier (only using the text content) to assign the pseudo labels for the other tables. At last, we manually clean all the labels. 5) Given the cells type information, The cells linking annotations are firstly generated by the heuristic rules that all values are displayed on the right or bottom of the keys. We also have to clean these annotations manually since rules usually fail in many complicated situations.

Table 1: Comparison with public table datasets. The top part contains the datasets based on the image for the table recognition task, and the bottom part are the datasets for table understanding tasks.

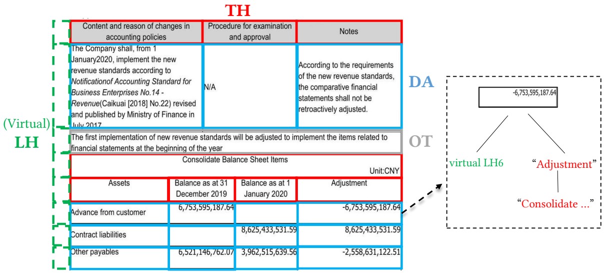
Dataset	Compound Table Ratio	Format	Amount	Language	Annotations					
					Cells Location	Textual Location	Textual Content	Columns/ Rows Indexes	Cells Type	Cells Linking
UNLV[50]	0	image	427	English		✓	✓	✓		
ICDAR2013[19]	0	image, pdf	150	English		✓	✓	✓		
ICDAR2019[16]	0	image	2,000	English	✓	✓		✓		
PubTabNet[74]	¡ 2 %	image	568,000	English		✓	✓	✓	✓	
SciTSR[8]	¡ 2 %	image, pdf	15,000	English		✓	✓	✓		
FinTabNet[72]	¡ 2 %	image, pdf	113,000	English		✓	✓	✓		
TableBank[32]	¡ 2 %	image	417,234	English				✓		
PubTables-1M[54]	0	image, pdf	1M	English	✓		✓	✓	✓	
WebSheet[12]	0	spreadsheet	3,503	English			✓	✓	✓	
SAUS[17]	0	spreadsheet	210	English			✓	✓	✓	
CIUS[17]	0	spreadsheet	268	English			✓	✓	✓	
DeEX[29]	0	excel	216	English			✓	✓	✓	
Spider[69]	0	json, sqlite	200	English			✓	✓	✓	
WikiSQL[73]	0	json, db	24,241	English			✓	✓	✓	
WikiTableQuestions[43]	0	csv, html	2,108	English			✓	✓	✓	
HybridQA[6]	0	csv, html	13,000	English			✓	✓	✓	✓
TabFact[5]	0	csv, html	16,573	English			✓	✓	✓	✓
ComFinTab	~70 %	image	10,000	English & Chinese	✓	✓	✓	✓	✓	✓

3.2 Uniform Table Understanding Task

The target of visual table understanding is to extract useful information. Previous works defined different table understanding tasks, such as table type classification [9, 14, 30, 62], cell type classification [17, 20, 58, 62], cell relation linking [27], Table QA [25, 43, 68]. Without considering the complicated questions involving logical reasoning in Table QA, we attempt to find a uniform task target that can be used in the proposed compound table dataset and previous basic table datasets.

Referring to the existing taxonomies of cell types [12, 17, 29, 62], we define cells into four general categories: Top Header (TH), Left Header(LH), Data (DA), and Other(OT), as shown in Figure 2. In a table, the headers TH and LH serve as the key to lead out a group of items, and DA is the value that stores the specific information. The TH and LH can be organized in hierarchical ways. We simply ignore the cells that cannot be categorized into TH/LH/DA and define them as OT, which might provide additional information like table title, metadata, comments, *etc.* In a 2D table, we have to use the keys in both TH and LH to locate a piece of information uniquely. Noticing that tables might lack one of the heads, we define the virtual heads that indicate the row/column indexes. This definition actually fits the habit of people reading tables. For example, in the bottom of the sub-table (which can be treated as a relational table) shown in Figure 2, to uniquely identify the value of "-6753595187.64", we might ask the question like " What is the *Consolidate Balance Sheet Items*'s *Adjustment*'s value in 6-th row?" In this way, cells with the same virtual left head can be correctly associated.

Given the cells categories, we can define the uniform target to find all the *table items* for a given table image, named table items extraction. A table item can be represented by a tree where the root node is the specific DA cell, and the left and right sub-tree store the LH and TH information with hierarchy, respectively. The leaf nodes denote the leftmost or topmost heads in a basic table. If there is no LH or TH, we use virtual heads to represent it, as an example illustrated in Figure 2. The form of *table item* contains information on cell types and relations, which can be transferred into almost all the results of the previous task forms designed for basic tables.



Content and reason of changes in accounting policies	Procedure for examination and approval	Notes
The Company shall, from 1 January2020, implement the new revenue standards according to *Notificationof Accounting Standard for Business Enterprises No.14 - Revenue*(Caikuai [2018] No.22) revised and published by Ministry of Finance in July 2017	N/A	According to the requirements of the new revenue standards, the comparative financial statements shall not be retroactively adjusted.

The first implementation of new revenue standards will be adjusted to implement the items related to financial statements at the beginning of the year

Consolidate Balance Sheet Items

Unit:CNY

Assets	Balance as at 31 December 2019	Balance as at 1 January 2020	Adjustment
Advance from customer	6,753,595,187.64		-6,753,595,187.64
Contract liabilities		8,625,433,531.59	8,625,433,531.59
Other payables	6,521,146,762.07	3,962,515,639.56	-2,558,631,122.51

Figure 2: An example of cells type (left) and a basic table item represented by a tree (right).

Evaluation Metric. Based on the task objective, the performance can be simply defined as the F1-Score of all the predicted table items compared to the ground truth. Nevertheless, predictions can be flawed to varying degrees. Inspired by [74], we use Tree-Edit-Distance-Based Similarity (TEDS) to denote the matching degree of two tables. So, the new metric of *Tree-F1-Score* can be calculated as:

$$\textit{Tree-R} = \frac{\sum_{t_i \in \mathcal{T}_G} \mathrm{TEDS}(t_i, t_i')}{||\mathcal{T}_G||} \tag{1}$$

$$\textit{Tree-P} = \frac{\sum_{t_i' \in \mathcal{T}_P} \mathrm{TEDS}(t_i', t_i)}{||\mathcal{T}_P||} \tag{2}$$

$$\textit{Tree-F1} = \frac{2 \times \textit{Tree-R} \times \textit{Tree-R}}{\textit{Tree-R} + \textit{Tree-P}} \tag{3}$$

where $\mathcal{T}_P$ and $\mathcal{T}_G$ denote the predicted and ground-truth tree set, respectively. t'_i is the item in $\mathcal{T}_P$ having the same root with t_i in $\mathcal{T}_G$. If the corresponding node cannot be found in the counterpart set, the TEDS is 0.

4 COMPOUND TABLE UNDERSTANDING MODELING

Based on the task form mentioned above, we propose a novel framework that supports both table recognition and understanding tasks, named Compound Table Understanding Network (CTUNet). The overview of CTUNet is shown in Figure 3, which can be separated into four parts: (1) The Table Element Extraction part aims to obtain the cell's location, text's location and content. (2) The Structural Graph Construction process sets up the spatial relationships between cells for the table according to the cell's location. (3) A Multi-modal Feature Fusion module integrates the visual, textual, and position features extracted from the previous Table Element Extraction module. (4) The final Relational Graph Construction module predicts the semantic relations among cells, integrating a Masked Self-Attention module to enhance the graph feature in each node. It is in charge of outputting the definitive collection of table items. The overall framework is trained end-to-end (excludes the OCR part), which makes the table recognition and understanding tasks benefit each other to a large extent.

4.1 Table Elements Extraction and Structural Graph Construction

The Table Elements Extraction (TEE) and Structural Graph Construction (SGC) modules together implement a basic table recognition process. Similar to previous works [8, 38, 46, 49, 51], the text content and structure information are obtained separately.

For the structure information, we first train a Faster-RCNN [48] model to detect the location of the cells, which is different from the text location. Cells contain the direct alignment information to construct the spatial relationships in rows and columns. Since all the tables are extracted from PDF files and all horizontally aligned, we adopt the post-processing of cell matching in [46] to construct the cells relations. Specifically, the criterion for connecting cells can be summarized as follows: If the two cells' bounding boxes have more than 50% overlapping in the x dimension, they will be connected in a vertical direction. A similar process can be conducted in the horizontal direction.

We adopt the offline OCR engine to extract the text location and content. Since a cell may contain multiple lines of text, we conduct a box-matching process to match the text's bounding boxes with the cells and combine the text contents that belong to the same cell. Specifically, we will match the text bounding boxes with the cell box with the highest Intersect-Over-Union (IOU) and combine the textual content from top to bottom according to the y coordinate.

4.2 Multi-Modal Feature Fusion

To fully understand a table, the information provided by a single modal is somehow limited. Similar to most previous work to understand document[56, 64, 65, 70], we summarize useful information into three feature modalities: position feature, visual feature, and textual feature.

Position Feature. Given the predicted cells $\mathcal{C}=\{c_1, c_2, ..., c_N\}$ where N is the number of cells, we denote the bounding boxes for cell c_i as $b_i=(x_1, y_1, x_2, y_2)$. The position feature is introduced by embedding the positions into a feature sequence, which can be formed as $PE_i=embedding(b_i)$, where $PE_i \in \mathbb{R}^{d_F}$ and d_F is the feature dimension.

Visual Feature. We conduct the RoI-Align operation to crop the detected cell regions from the high-level feature map of the cell detection network. All the feature maps will be resized into a fixed shape, denoted as $\{f_1, f_2, ...\}$. The final visual feature map for i-th cell can be obtained as $V_i=Linear(Convs(f_i))$, where $Convs()$ is a stack of convolution layers and $Linear()$ operation transfers the feature into the shape of $\mathbb{R}^{d_F}$.

Textual features. The textual feature contains the underlying habits of people using language, especially in the financial field. To dig out the patterns in the language and help in improving the table understanding, we adopt BERT [10] to extract the textual content of the cell (BERT-Chinese for ComFinTab-Chinese) and obtain the final textual feature embedding, denoted as $T_i \in \mathbb{R}^{d_F}$.

To support the following relation prediction, the initial feature of the i-th cell is generated by concatenating all three features and then normalizing it as follows,

$$F_i = LayerNorm(PE_i + V_i + T_i). \tag{4}$$

4.3 Relational Graph Construction

In the Relational Graph Construction (RGC) module, we treat each cell as a node in a graph. The task of table items extraction can be transferred into predicting the nodes' classification and the existing edges between nodes.

To ensure that each 'DA' node has corresponding edges in both directions, we will create the virtual 'LH' or 'TH' nodes when some sub-tables are missing a certain type of head. The virtual nodes are generated according to the maximum column/row numbers from the structural information provided by the previous SGC module. The positions of the virtual nodes will be assigned as the small empty regions extended from the left-most or top-most cells, as shown in Figure 2. The visual and textual features will be filled with default values.

Given the fused feature of each node, we adopt the masked self-attention mechanism in the graph attention network [60] to further enhance the node features by aggregating the neighbor's information. Specifically, for the i-th node, the enhanced feature F'_i can be calculated as follows,

$$e_{ij} = \text{LeakyReLU}\left(w^T \left[WF_i \middle|\middle| WF_j\right]\right) \tag{5}$$

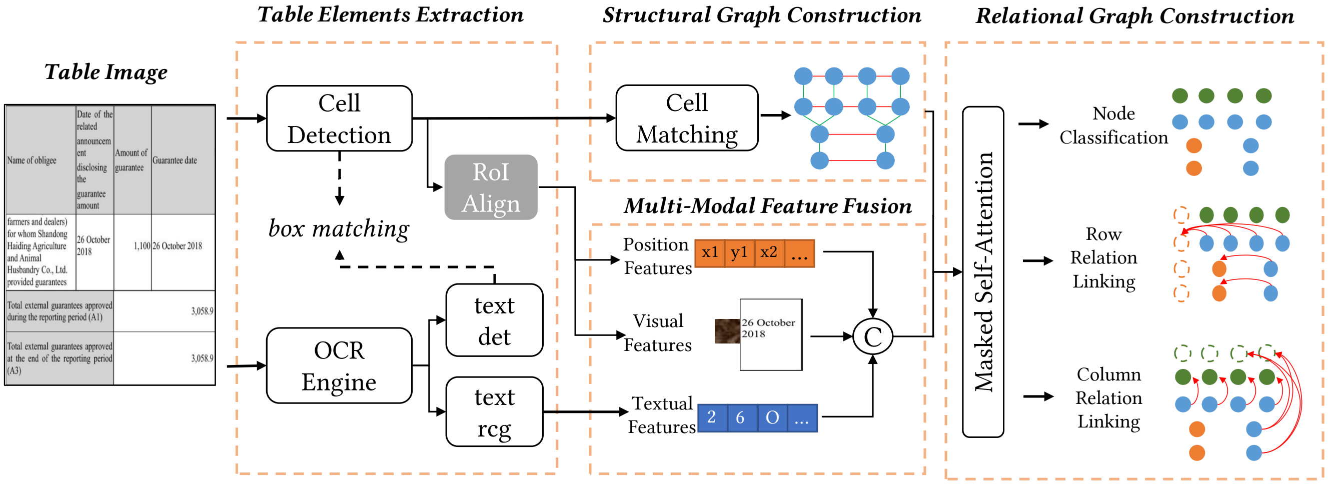


Table Image
Name of obligee
Date of the related announcement disclosing the guarantee amount
Amount of guarantee
Guarantee date
farmers and dealers) for whom Shandong Haiding Agriculture and Animal Husbandry Co., Ltd. provided guarantees
26 October 2018
1,100
26 October 2018
Total external guarantees approved during the reporting period (A1)
3,058.9
Total external guarantees approved at the end of the reporting period (A3)
3,058.9
Table Elements Extraction
Cell Detection
box matching
RoI Align
OCR Engine
text det
text rcg
Structural Graph Construction
Cell Matching
Multi-Modal Feature Fusion
Position Features
x1 y1 x2 ...
Visual Features
26 October 2018
Textual Features
2 6 O ...
C
Relational Graph Construction
Masked Self-Attention
Node Classification
Row Relation Linking
Column Relation Linking

Figure 3: Overview of the CTUNet framework. It contains (1) a Table Element Extraction module to extract the basic elements in a table, including the cell location, text location and content; (2) a Structural Graph Construction module used to infer the structure of the table; (3) a Multi-modal Feature Fusion module that integrates the visual, textual and position features to obtain a comprehensive feature embedding; (4) the Relational Graph Construction module to predict the semantic relations among cells and outputs the final collection of table items.

$$\alpha_{ij} = \mathrm{softmax}_j(e_{ij}) = \frac{exp(e_{ij})}{\sum_{k\in\mathcal{N}_i} exp(e_{ik})} \tag{6}$$

$$F_i^{'} = \sigma\left(\sum_{j\in\mathcal{N}_i} \alpha_{ij} W F_j\right) \tag{7}$$

where $[.||.]$ is the concatenation operation, w is a single-layer feed-forward neural network transformation, W is the weight matrix, and $\mathcal{N}_i$ is the collection of i-th node's neighborhood in the graph. $F_i' \in \mathcal{F}$ is the enhanced node feature with shape of $\mathbb{R}^{d_F}$. Expressly, we set the neighborhood of node i as all nodes that are connected with it in the structural graph generated in the SGC module, *i.e.*, the nodes having the same row or column with node i. This setting is based on the prior knowledge that if two nodes are not connected structurally, they must not be connected semantically.

After getting the aggregated features for each node, we concatenate the features of each cell pairwise to get the edges features:

$$\mathcal{E} = \begin{bmatrix} E_{1,1} & \cdots & E_{1,m} \\ \vdots & \ddots & \vdots \\ E_{m,1} & \cdots & E_{m,m} \end{bmatrix} \tag{8}$$

$$E_{i,j} = [F_i'||F_j'] \tag{9}$$

where m is the total node number, $E_{i,j} \in \mathbb{R}^{2d_F}$ is the edge feature between node i and j and $\mathcal{E} \in \mathbb{R}^{m^2 \times 2d_F}$.

Then, the RGC module will be simultaneously trained with three tasks, a node classification task and the edge classification tasks separately in horizontal and vertical directions, named Row Relation Linking and Column Relation Linking. Specifically, based on the node features $\mathcal{F}$ and edge feature $\mathcal{E}$, the final prediction of three branches can be formed as,

$$P_{node} = \mathrm{softmax}(FC(\mathcal{F})), \tag{10}$$

$$P_{rlink} = \mathrm{sigmoid}(FC(\mathcal{E})), \tag{11}$$

$$P_{clink} = \mathrm{sigmoid}(FC(\mathcal{E})), \tag{12}$$

where $FC()$ is a fully-connected layer, P_{node}, P_{clink} and P_{clink} denote the predictions of node classification, column relation linking and row relation linking, respectively.

In the post-processing, we will combine the classification results of the nodes and edges to form the definitive collection of table items, where we only consider the cells that belong to the 'DA' type. Notice that although we can also generate the virtual nodes and links in the post-processing according to the tables type, involving the virtual nodes in the graph's feature modeling can effectively prevent cells from being connected with the fake heads.

Optimization. In addition to the OCR engine used to extract the textual contents and locations, the framework can be trained end-to-end. The overall loss can be formed as,

$$L = L_{det} + \lambda_1 L_{node} + \lambda_2 (L_{rlink} + L_{clink}), \tag{13}$$

where L_{det} is the loss for cell detection network and L_{node}, L_{rlink}, L_{clink} are the cross entropy losses that used in the node classification and two edge classification tasks.

5 EXPERIMENT

5.1 Experimental Settings

Implementation Details. The backbone of our model is a 50-layer ResNet [24], followed by the FPN [36] to further enhance features. For all benchmarks, the model is trained by the SGD optimizer with *batch size*= 4, *momentum*=0.9, and *weight decay*= 1×10^{-4}. The initial learning ratio is 1×10^{-3}, which is then divided by 10 every 20 epochs. All experiments are implemented in Pytorch with 8 Tesla-V100 GPUs.

We split the training / testing set as 4500/1500 images for the ComFinTab-Chinese dataset and 3200/800 images for ComFinTab-English data, respectively. Because the table layout from the same company may be similar to each other, to ensure that the model does not overfit into certain formats, we follow [72] to separate the dataset at a company level.

Compared Methods. Under the new task form mentioned above, no previous work can be directly compared. So we set up the following two compared experimental settings.

(1) Cell Classification + Rules. Previous works on table understanding provide support for cell-type classification. So we select one of the current advanced methods, TUTA [62], which is a pre-trained language model-based method designed for tables. We fine-tune the model on the task of cell classification and then use carefully designed handcraft rules to achieve cell relation linking. The basic idea of the rule is to match the header cells with the cells in their right/bottom of the same rows/columns. Since TUTA is an English pre-trained model, we only report its result on ComFinTab-English. Moreover, we also compare another SOTA document understanding approach for the cell classification task using multi-modal information, LayoutLM v2[65].

(2) End-to-End Relation Extraction. To reduce the error impact of manual rules, we consider an alternative end-to-end approach. We find that the output of the relation extraction task in NLP, usually in a triple form like ('Entity A', 'Relation', 'Entity B'), can represent a cell link status in the table to a large extent. Specifically, for a table item represented by a tree, we convert it into several triples like ('DA 1', 'Left-Linking', 'LH 1'), ('DA 1', 'Top-Linking', 'TH 1'), where 'Left-Linking' and 'Top-Linking' are pre-defined two relation types. We can make the ground truth for the hierarchical head as ('LH 1', 'Left-Linking', 'LH 2'). In this way, we can build the model with an end-to-end relation extraction task target where each node can be treated as a token in sequence. Here, we compare our model with one of the advanced methods AGCCN [22]. We also introduce the multi-modal features in AGCCN for a fair comparison.

All the methods use the same textual information extracted by the OCR engines, *i.e.*, PaddleOCR [13] for ComFinTab-Chinese and Tesseract [53] for ComFinTab-English. To eliminate the impact of errors in OCR and structural recognition to evaluate the pure table understanding ability, we also report the result of the model using the ground truths of OCR and cell locations, denoted as 'Ours(GT)'.

5.2 Results

The experimental results are shown in Table 2. We also report the table recognition result of our method as a reference using the metric of TEDS [74]. The TEDS results show that the current data set is relatively simple for the table recognition task since most tables have visible grid lines.

We separately report the performance of the understanding task using metrics of the *Macro-F1 score* for cell-type classification and the proposed *Tree-F1-Score* for table item extraction. We can see that the differences in performance of the cell classification task between models are relatively small. Although TUTA achieves the best F1 score in this task, it has been pre-trained on large table datasets. For the performance of table item extraction, although the rule-based settings (TUTA+rule, LayoutLM v2+rule) obtain satisfied cell classification results, they are easy to be failed in the relation extraction in the compound table datasets. In contrast, the end-to-end setting of AGCCN can eliminate the error generated by the heuristic rules to some extent. Our proposed model is also an end-to-end framework which makes full use of the table structural information. It can transmit the multi-modal feature among cells and predict the relations in a uniform and intuitional task target, resulting in better results. The above experimental phenomena remain consistent when we use the ground truths of OCR and cells.

Figure 4 has shown some visualized images. Different images might need different clues to understand their compound structure. For example, in the first image, the background color of cells provides a clear signal for cell-type classification. However, in the second image, the cell type and their relations should be more inferred from the layout and semantic information. The third image shows a failure case in that our model misjudges the type of cell in the second row. It is because the background color information somehow misleads the model. Nevertheless, our model shows strong ability and adaptability to compound tables from the results demonstrated above.

5.3 Ablation Studies

In order to better observe the contribution of our proposed modules to the table understanding task, the following ablation experiment is based on ground truths of OCR and cell location.

Ablation on Multi-Modal Features. We conduct the ablation experiments to verify the contributions of the multi-modal features, whose results are shown in Table 3. The results show that if there is no position feature, the accuracy will be reduced dramatically. It is because the table itself is a highly structured data form, where the alignment information is one of the fundamental characteristics. Furthermore, for both visual and semantic features, any one of them can improve the model's performance to some extent because they can add much important extra information for understanding. The model can achieve the best performance when integrating all three modality information. Moreover, for the visual and semantic feature, no matter which modality is missing, there will be at least a 5% performance drop in the accuracy for the table item extraction task. However, for the cell type classification task, such decant was only about 1.5%. It shows

Table 2: Table item extraction results on ComFinTab-English and ComFinTab-Chinese. Methods labelled with (GT) means to use the ground truths of OCR and cells location.

Methods	ComFinTab-English					ComFinTab-Chinese				
	TEDS	Cell-F1	Tree-P	Tree-R	Tree-F1	TEDS	Cell-F1	Tree-P	Tree-R	Tree-F1
TUTA [62] + rule	-	**92.38**	72.68	73.14	72.91	-	-	-	-	-
LayoutLM v2 [65] + rule	-	91.93	72.03	72.07	72.05	-	91.67	71.72	71.66	71.69
AGCCN [22]	-	91.64	85.23	84.11	84.67	-	91.38	84.98	83.31	84.14
Ours	98.99	92.27	**89.02**	**89.38**	**89.20**	98.83	**91.78**	**87.11**	**86.86**	**86.98**
TUTA (GT) [62] + rule	-	**93.01**	74.12	74.05	74.08	-	-	-	-	-
LayoutLM v2 (GT) [65] + rule	-	92.72	72.99	73.85	73.42	-	92.14	73.73	73.87	73.80
AGCCN (GT) [22]	-	92.50	87.16	85.21	86.17	-	91.92	87.28	84.92	86.08
Ours(GT)	99.93	92.98	**90.45**	**90.30**	**90.37**	99.89	**92.45**	**89.25**	**88.55**	**88.90**

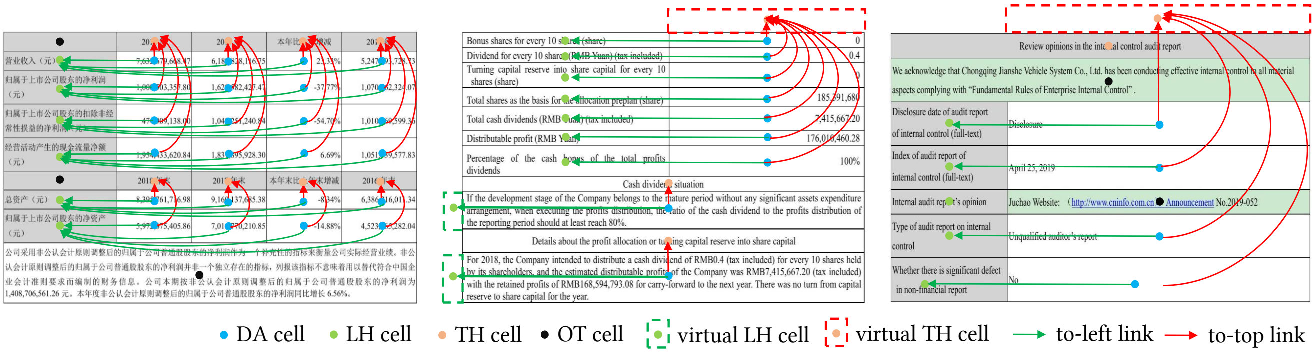
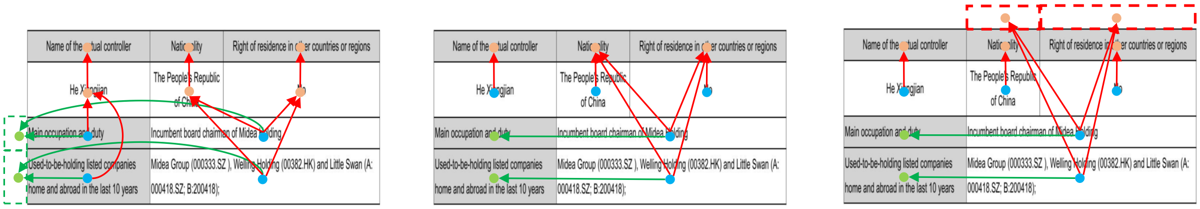


Figure 4: Visualization of some samples in ComFinTab with the predicted cell types and cell links. Better viewed in color.



(a) position

(b) position + visual

(c) position + visual + semantic

Figure 5: Some visualization examples that use different modality features. Better viewed in color.

that table item extraction is a more challenging task requiring tight integration of model information.

Figure 5 shows a visualized comparison that use different modality information. If we only use the position information, the model predicts only based on the alignment information, which easily misjudges the cell type. After adding the visual feature, the model can obtain the visual features like color to predict better. However, the relationship predictions are still wrong since some aligned cells actually belong to different sub-tables. Finally, we can obtain the exact result when the model integrates with all three modalities.

Ablation on Relational Graph Construction Module. In the RGC module, we adopt the masked self-attention mechanism in the graph attention network to enhance the node and edge features. During the node feature updating in the graph attention network, a node will aggregate the features from all its neighbors. So choosing different neighbors will have different results. Table 4 demonstrates the results on using different node neighborhood settings. In the table,

Table 3: The results of the ablation on different modality features. (E) means ComFinTab-English and (C) means ComFinTab-Chinese.

Position	-	✓	✓	✓	✓
Visual	✓	-	✓	-	✓
Semantic	✓	-	-	✓	✓
Cell-F1 (E)	89.23	87.80	91.35	91.55	**92.98**
Tree-P (E)	50.33	79.91	86.25	85.63	**90.45**
Tree-R (E)	30.44	78.53	85.40	84.10	**90.30**
Tree-F1 (E)	37.93	79.21	85.82	84.68	**90.37**
Cell-F1 (C)	87.93	86.55	91.28	91.49	**92.45**
Tree-P (C)	37.68	79.14	85.33	84.60	**89.25**
Tree-R (C)	26.81	78.05	84.48	83.68	**88.55**
Tree-F1 (C)	31.32	77.56	84.90	84.13	**88.90**

Table 4: Results on different node neighbors in the structural graph.

Neighbor	ComFinTab-English		ComFinTab-Chinese	
Settings	Cell-F1	Tree-F1	Cell-F1	Tree-F1
None	91.70	88.19	91.71	87.64
1-step	92.10	89.35	92.04	88.46
All	92.22	89.66	92.08	88.53
Row/Column	**92.98**	**90.37**	**92.45**	**88.90**

'None' means without doing any feature aggregation, 'All' means aggregating all other nodes' features in a complete graph, '1-step' considers the cells spatially direct adjacent to the current cell as neighbors, and 'Row/Column' is our model's setting, which regards the cells belong to the same row/columns as neighbors. Comparing 'None' result and the others, we can easily see the effectiveness of the self-attention operation. No matter what kind of neighbor node is used, the introduction of self-attention can improve the accuracy. Among them, our setting achieves the best performance since it involves strong prior knowledge about the correspondence of the rows and columns.

6 CONCLUSION

This paper establishes a new image-based dataset, ComFinTab, with rich annotations for the table recognition and table understanding task, including many challenging compound tables. In order to better adapt to the compound tables, a concise and uniform table understanding task with the evaluation metric is proposed, which can also be applied to almost all previous datasets. Finally, we propose a novel table understanding framework, named CTUNet, for this dataset that makes full use of visual, semantic, and position features. CTUNet integrates a graph-attention network to further enhance the features and train with two table understanding tasks end-to-end. The experimental results demonstrate the effectiveness and robustness of our proposed model.

REFERENCES

[1] Eva Banik, Claire Gardent, and Eric Kow. 2013. The KBGen Challenge. In *ENLG*. 94–97.

[2] David L. Chen and Raymond J. Mooney. 2008. Learning to sportscast: a test of grounded language acquisition. In *ICML*, Vol. 307. 128–135.

[3] Jiaoyan Chen, Ernesto Jiménez-Ruiz, Ian Horrocks, and Charles Sutton. 2019. ColNet: Embedding the Semantics of Web Tables for Column Type Prediction. In *AAAI*. 29–36.

[4] Wenhu Chen, Jianshu Chen, Yu Su, Zhiyu Chen, and William Yang Wang. 2020. Logical Natural Language Generation from Open-Domain Tables. In *ACL*. 7929–7942.

[5] Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020. TabFact: A Large-scale Dataset for Table-based Fact Verification. In *ICLR*.

[6] Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. 2020. HybridQA: A Dataset of Multi-Hop Question Answering over Tabular and Textual Data. In *EMNLP*. 1026–1036.

[7] Zhoujun Cheng, Haoyu Dong, Zhiruo Wang, Ran Jia, Jiaqi Guo, Yan Gao, Shi Han, Jian-Guang Lou, and Dongmei Zhang. 2021. HiTab: A Hierarchical Table Dataset for Question Answering and Natural Language Generation. *CoRR* abs/2108.06712 (2021).

[8] Zewen Chi, Heyan Huang, Heng-Da Xu, Houjin Yu, Wanxuan Yin, and Xianling Mao. 2019. Complicated Table Structure Recognition. *CoRR* abs/1908.04729 (2019).

[9] Eric Crestan and Patrick Pantel. 2011. Web-scale table census and classification. In *WSDM*. 545–554.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*. 4171–4186.

[11] Haoyu Dong, Jinyu Wang, Zhouyu Fu, Shi Han, and Dongmei Zhang. 2020. Neural Formatting for Spreadsheet Tables. In *CIKM*. 305–314.

[12] Haoyu Dong, Jiong Yang, Shi Han, and Dongmei Zhang. 2020. Learning Formatting Style Transfer and Structure Extraction for Spreadsheet Tables with a Hybrid Neural Network Architecture. In *CIKM*. 2389–2396.

[13] Yuning Du, Chenxia Li, Ruoyu Guo, Xiaoting Yin, Weiwei Liu, Jun Zhou, Yifan Bai, Zilin Yu, Yehua Yang, Qingqing Dang, and Haoshuang Wang. 2020. PP-OCR: A Practical Ultra Lightweight OCR System. *CoRR* abs/2009.09941 (2020).

[14] Julian Eberius, Katrin Braunschweig, Markus Hentsch, Maik Thiele, Ahmad Ahmadov, and Wolfgang Lehner. 2015. Building the Dresden Web Table Corpus: A Classification Approach. In *BDC*. 41–50.

[15] Jing Fang, Xin Tao, Zhi Tang, Ruiheng Qiu, and Ying Liu. 2012. Dataset, Ground-Truth and Performance Metrics for Table Detection Evaluation. In *DAS*. 445–449.

[16] Liangcai Gao, Yilun Huang, Hervé Déjean, Jean-Luc Meunier, Qinqin Yan, Yu Fang, Florian Kleber, and Eva Maria Lang. 2019. ICDAR 2019 Competition on Table Detection and Recognition (cTDaR). In *ICDAR*. 1510–1515.

[17] Majid Ghasemi-Gol, Jay Pujara, and Pedro A. Szekely. 2019. Tabular Cell Classification Using Pre-Trained Cell Embeddings. In *ICDM*. 230–239.

[18] Majid Ghasemi-Gol and Pedro A. Szekely. 2018. TabVec: Table Vectors for Classification of Web Tables. *CoRR* abs/1802.06290 (2018).

[19] Max C. Göbel, Tamir Hassan, Ermelinda Oro, and Giorgio Orsi. 2013. ICDAR 2013 Table Competition. In *ICDAR*. 1449–1453.

[20] Julius Gonsior, Josephine Rehak, Maik Thiele, Elvis Koci, Michael Günther, and Wolfgang Lehner. 2020. Active Learning for Spreadsheet Cell Classification. In *EDBT/ICDT (CEUR Workshop Proceedings, Vol. 2578)*.

[21] Tong Guo, Derong Shen, Tiezheng Nie, and Yue Kou. 2020. Web Table Column Type Detection Using Deep Learning and Probability Graph Model. In *WISA*. 401–414.

[22] Zhijiang Guo, Yan Zhang, and Wei Lu. 2019. Attention Guided Graph Convolutional Networks for Relation Extraction. In *ACL*. 241–251.

[23] Khurram Azeem Hashmi, Marcus Liwicki, Didier Stricker, Muhammad Adnan Afzal, Muhammad Ahtsham Afzal, and Muhammad Zeshan Afzal. 2021. Current Status and Performance Analysis of Table Recognition in Document Images With Deep Neural

Networks. *IEEE Access* 9 (2021), 87663–87685.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *CVPR*. 770–778.

[25] Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Martin Eisenschlos. 2020. TaPas: Weakly Supervised Table Parsing via Pre-training. In *ACL*. 4320–4333.

[26] Yilun Huang, Qinqin Yan, Yibo Li, Yifan Chen, Xiong Wang, Liangcai Gao, and Zhi Tang. 2019. A YOLO-Based Table Detection Method. In *ICDAR*. 813–818.

[27] Guillaume Jaume, Hazim Kemal Ekenel, and Jean-Philippe Thiran. 2019. FUNSD: A Dataset for Form Understanding in Noisy Scanned Documents. In *OST@ICDAR*. 1–6.

[28] Elvis Koci, Maik Thiele, Wolfgang Lehner, and Oscar Romero. 2018. Table Recognition in Spreadsheets via a Graph Representation. In *DAS*. 139–144.

[29] Elvis Koci, Maik Thiele, Josephine Rehak, Oscar Romero, and Wolfgang Lehner. 2019. DECO: A Dataset of Annotated Spreadsheets for Layout and Table Recognition. In *ICDAR*. 1280–1285.

[30] Larissa R. Lautert, Marcelo M. Scheidt, and Carina F. Dorneles. 2013. Web table taxonomy and formalization. *SIGMOD Rec.* 42, 3 (2013), 28–33.

[31] Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural Text Generation from Structured Data with Application to the Biography Domain. In *EMNLP*. 1203–1213.

[32] Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, Ming Zhou, and Zhoujun Li. 2020. TableBank: Table Benchmark for Image-based Table Detection and Recognition. In *LREC*. 1918–1925.

[33] Yibo Li, Liangcai Gao, Zhi Tang, Qinqin Yan, and Yilun Huang. 2019. A GAN-Based Feature Generator for Table Detection. In *ICDAR*. 763–768.

[34] Yiren Li, Zheng Huang, Junchi Yan, Yi Zhou, Fan Ye, and Xianhui Liu. 2020. GFTE: Graph-Based Financial Table Extraction. In *ICPR*. 644–658.

[35] Percy Liang, Michael I. Jordan, and Dan Klein. 2009. Learning Semantic Correspondences with Less Supervision. In *ACL*. 91–99.

[36] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, B-harath Hariharan, and Serge J. Belongie. 2017. Feature Pyramid Networks for Object Detection. In *CVPR*. 936–944.

[37] Hao Liu, Xin Li, Bing Liu, Deqiang Jiang, Yinsong Liu, Bo Ren, and Rongrong Ji. 2021. Show, Read and Reason: Table Structure Recognition with Flexible Context Aggregator. In *ACM Multimedia*. 1084–1092.

[38] Rujiao Long, Wen Wang, Nan Xue, Feiyu Gao, Zhibo Yang, Y-ongpan Wang, and Gui-Song Xia. 2021. Parsing Table Structures in the Wild. In *ICCV*. 924–932.

[39] Kyosuke Nishida, Kugatsu Sadamitsu, Ryuichiro Higashinaka, and Yoshihiro Matsuo. 2017. Understanding the Semantic Structures of Tables with a Hybrid Deep Neural Network Architecture. In *AAAI*. 168–174.

[40] Jekaterina Novikova, Oliver Lemon, and Verena Rieser. 2016. Crowd-sourcing NLG Data: Pictures Elicit Better Data. In *INLG*. 265–273.

[41] Shubham Singh Paliwal, Vishwanath D, Rohit Rahul, Monika Sharma, and Lovekesh Vig. 2019. TableNet: Deep Learning Model for End-to-end Table Detection and Tabular Data Extraction from Scanned Document Images. In *ICDAR*. 128–133.

[42] Ankur P. Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. ToTTo: A Controlled Table-To-Text Generation Dataset. In *EMNLP*. 1173–1186.

[43] Panupong Pasupat and Percy Liang. 2015. Compositional Semantic Parsing on Semi-Structured Tables. In *ACL*. 1470–1480.

[44] Devashish Prasad, Ayan Gadpal, Kshitij Kapadni, Manish Visave, and Kavita Sultanpure. 2020. CascadeTabNet: An approach for end to end table detection and structure recognition from image-based documents. In *CVPR Workshops*. 2439–2447.

[45] Shah Rukh Qasim, Hassan Mahmood, and Faisal Shafait. 2019. Rethinking Table Recognition using Graph Neural Networks. In *ICDAR*. 142–147.

[46] Liang Qiao, Zaisheng Li, Zhanzhan Cheng, Peng Zhang, Shiliang Pu, Yi Niu, Wenqi Ren, Wenming Tan, and Fei Wu. 2021. LGPMA: Complicated Table Structure Recognition with Local and Global Pyramid Mask Alignment. In *ICDAR*. 99–114.

[47] Sachin Raja, Ajoy Mondal, and C. V. Jawahar. 2020. Table Structure Recognition Using Top-Down and Bottom-Up Cues. In *ECCV*. 70–86.

[48] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NeurIPS*. 91–99.

[49] Sebastian Schreiber, Stefan Agne, Ivo Wolf, Andreas Dengel, and Sheraz Ahmed. 2017. DeepDeSRT: Deep Learning for Detection and Structure Recognition of Tables in Document Images. In *ICDAR*. 1162–1167.

[50] Asif Shahab, Faisal Shafait, Thomas Kieninger, and Andreas Dengel. 2010. An open approach towards the benchmarking of table structure recognition systems. In *DAS*, David S. Doermann, Venu Govindaraju, Daniel P. Lopresti, and Premkumar Natarajan (Eds.). 113–120.

[51] Shoaib Ahmed Siddiqui, Imran Ali Fateh, Syed Tahseen Raza Rizvi, Andreas Dengel, and Sheraz Ahmed. 2019. DeepTabStR: Deep Learning based Table Structure Recognition. In *ICDAR*. 1403–1409.

[52] Shoaib Ahmed Siddiqui, Pervaiz Iqbal Khan, Andreas Dengel, and Sheraz Ahmed. 2019. Rethinking Semantic Segmentation for Table Structure Recognition in Documents. In *ICDAR*. 1397–1402.

[53] R. Smith. 2007. An Overview of the Tesseract OCR Engine. In *ICDAR*. 629–633.

[54] Brandon Smock, Rohith Pesala, and Robin Abraham. 2021. PubTables-1M: Towards comprehensive table extraction from unstructured documents. *CoRR* abs/2110.00061 (2021).

[55] Lya Hulliyyatus Suadaa, Hidetaka Kamigaito, Kotaro Funakoshi, Manabu Okumura, and Hiroya Takamura. 2021. Towards Table-to-Text Generation with Numerical Reasoning. In *ACL/IJCNLP*. 1451–1465.

[56] Hongbin Sun, Zhanghui Kuang, Xiaoyu Yue, Chenhao Lin, and Wayne Zhang. 2021. Spatial Dual-Modality Graph Reasoning for Key Information Extraction. *CoRR* abs/2103.14470 (2021).

[57] Huan Sun, Hao Ma, Xiaodong He, Wen-tau Yih, Yu Su, and Xifeng Yan. 2016. Table Cell Search for Question Answering. In *WWW*. 771–782.

[58] Kexuan Sun, Harsha Rayudu, and Jay Pujara. 2021. A Hybrid Probabilistic Approach for Table Understanding. In *AAAI*. 4366–4374.

[59] Chris Tensmeyer, Vlad I. Morariu, Brian L. Price, Scott Cohen, and Tony R. Martinez. 2019. Deep Splitting and Merging for Table Structure Decomposition. In *ICDAR*. 114–121.

[60] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *ICLR*.

[61] Yushi Wang, Jonathan Berant, and Percy Liang. 2015. Building a Semantic Parser Overnight. In *ACL*. 1332–1342.

[62] Zhiruo Wang, Haoyu Dong, Ran Jia, Jia Li, Zhiyi Fu, Shi Han, and Dongmei Zhang. 2021. TUTA: Tree-based Transformers for Generally Structured Table Pre-training. In *KDD*. 1780–1790.

[63] Sam Wiseman, Stuart M. Shieber, and Alexander M. Rush. 2017. Challenges in Data-to-Document Generation. In *EMNLP*. 2253–2263.

[64] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020. LayoutLM: Pre-training of Text and Layout for Document Image Understanding. In *KDD*. 1192–1200.

[65] Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei A. F. Florêncio, Cha Zhang, Wanxiang Che, Min Zhang, and Lidong Zhou. 2021. LayoutLMv2: Multimodal Pre-training for Visually-rich Document Understanding. In *ACL/IJCNLP*. 2579–2591.

[66] Wenyuan Xue, Baosheng Yu, Wen Wang, Dacheng Tao, and Qingyong Li. 2021. TGRNet: A Table Graph Reconstruction Network for Table Structure Recognition. In *ICCV*. 1275–1284.

[67] Jiaquan Ye, Xianbiao Qi, Yelin He, Yihao Chen, Dengyi Gu, Peng Gao, and Rong Xiao. 2021. PingAn-VCGroup's Solution for ICDAR 2021 Competition on Scientific Literature Parsing Task B: Table Recognition to HTML. *CoRR* abs/2105.01848 (2021).

[68] Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. TaBERT: Pretraining for Joint Understanding of Textual and Tabular Data. In *ACL*. 8413–8426.

[69] Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir R. Radev. 2018. Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task. In *EMNLP*. 3911–3921.

[70] Peng Zhang, Yunlu Xu, Zhanzhan Cheng, Shiliang Pu, Jing Lu, Liang Qiao, Yi Niu, and Fei Wu. 2020. TRIE: End-to-End Text Reading and Information Extraction for Document Understanding. In *ACM MM*. 1413–1422.

[71] Zhenrong Zhang, Jianshu Zhang, Jun Du, and Fengren Wang. 2022. Split, Embed and Merge: An accurate table structure recognizer. *Pattern Recognition.* 126 (2022), 108565.

[72] Xinyi Zheng, Douglas Burdick, Lucian Popa, Xu Zhong, and Nancy Xin Ru Wang. 2021. Global Table Extractor (GTE): A Framework for Joint Table Identification and Cell Structure Recognition Using Visual Context. In *WACV*. 697–706.

[73] Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning. *CoRR* abs/1709.00103 (2017).

[74] Xu Zhong, Elaheh ShafieiBavani, and Antonio Jimeno-Yepes. 2020. Image-Based Table Recognition: Data, Model, and Evaluation. In *ECCV*. 564–580.