Layout-Aware Neural Model for Resolving Hierarchical Table Structure

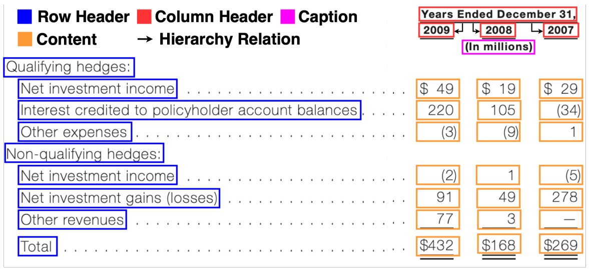
Anonymous ACL submission

Abstract

While many pipelines for extracting information from tables assume simple table structure, tables in the financial domain frequently have complex, hierarchical structure. The main example would be parent-child relationships between header cells. Most prior datasets of tables annotated from images or .pdf and most models for extracting table structure concentrate on the problems of table, cell, row, and column bounding box extraction. The area of fine-grained table structure remains relatively unexplored. In this study, we present a dataset of 887 tables, manually labeled for cell types and column hierarchy relations. The tables are selected from IBM FinTabNet, a much larger dataset of more than 100,000 financial tables having cell, row, and column bounding boxes extracted by deep learning, but not including semantic cell type or cell-to-cell relation labels, which we add. Selection of these 887 tables is performed using heuristics which result in a much larger proportion, roughly half, of the selected tables having complex hierarchical structure, than a random sample from FinTabNet. Further, we fine-tune models based on LayoutLM on the cell-type classification task and on the identification of hiearchical relations among column headers. We achieve F1 scores of 95% and 70% on the respective tasks. Finally, we use the trained model to create soft labels for the entirety of FinTabNet.

1 Introduction

Most work on automatic information extraction from tables assume that the table's structure is adequately represented by grouping of cells into simple rows and columns, in exactly the same way that the structure of a two-dimensional $m \times n$ array is represented by assigning each entry to a pair of integers $(i, j) \in [0, m-1] \times [0, n-1]$. In the case of tables found on the web, as in Wikipedia and related resources, for example, this assumption is largely borne out by experience. However, in some specialized domains, many of the tables do not have such a simple structure. In particular, in finance and financial reporting, there is an entrenched, culturally reinforced tendency to use rather complex table structure to convey information more concisely than a simple array-like table can. While such structures are intuitive to a human reader, they present an obstacle to the automation of information extraction from financial tables.


🟦 Row Header
🟥 Column Header
🟪 Caption
🟧 Content
→ Hierarchy Relation

	Years Ended December 31,		
	2009	2008	2007
		(In millions)	
Qualifying hedges:			
Net investment income	$ 49	$ 19	$ 29
Interest credited to policyholder account balances	220	105	(34)
Other expenses	(3)	(9)	1
Non-qualifying hedges:			
Net investment income	(2)	1	(5)
Net investment gains (losses)	91	49	278
Other revenues	77	3	—
Total	$432	$168	$269

Figure 1: Financial table annotated with fine structure

Fortunately, some analysis shows that the vast majority of deviations from simple table structure occurs in one of two main directions. The first is that the financial table has multiple layers of row or column headers, and there is a hierarchical tree-like structure to the row or column headers of the table. The second is that the table has text cells within the table that span multiple columns of mainly numerical cells. In analogy with the usual table *captions* which apply to the whole table, we can think of these cells as a special type of captions which apply only to a contiguous region of the table. In both cases, certain aspects of the table's structure that are not adequately captured by row-column assignments, can be represented by a directed tree structure. The nodes are row/column header cells (in the first case), or caption cells/content blocks (in the second case), and the edges correspond to the relation between two nodes that can be interpreted as "parent cell modifies or governs meaning of child cell". For example, in Figure 1, each of the three of the "child" column header cells ("2009", "2009", "2007") has its meaning modified or by the "parent" cell ("Years Ended December 31"). The caption "(in millions)" modifies the meaning of the "child" block of content cells outlined in yellow. In making these definition, we are simply rephrasing an observation made previously in, e.g., (Chen et al., 2017) and (Xue et al., 2019).

The main contributions of this work are as follows.

- We decompose the task of understanding the table structure, understood as identifying the correct tree structure as just outlined, as two simpler tasks. The first is a classification of all the cells in the table into four semantic classes, with labels *content*, *row header*, *column header*, *caption*, where "*caption*" is understood in the extended sense above. The second is a classification of all the potential relationship edges, as identified from all possible edges by some simple heuristics, into true/existing and false/non-existing relationship edges.
- We address both problems within a unified deep learning framework, namely the one provided by (Xu et al., 2020), which allows us to take advantage of the representations incorporating both semantic content of the cells and their surroundings and visual cues from the layout of the document.
- We produced and plan to release two datasets. The first is manually labelled with almost 900 tables, roughly half of which have complex structure. The second is a much larger dataset of 100K financial tables which are "soft-labeled" using a LayoutLM-based (Xu et al., 2020) model fine-tuned on the first dataset.

Because the data annotation procedures and protocols are a central part of our contribution, we devote an entire section, 3 below, to describing the dataset creation process in detail. Since row hierarchy structure tends to be more subjective than column hierarchy structure, we labelled only column header hierarchy. We intend to label row-header hierarchy in a future version, an effort which will require more resource-intensive review and resolution of inter-annotator disagreement. Despite this limitation, our manually labeled dataset of almost 900 tables is much larger than the typical dataset in this field (cf. (Chen et al., 2017) with 72 labeled examples, and no column hierarchy, only row-hierarchy).

Since tables that we are targeting for structure understanding are primarily in .pdf format, including in images, as found in the wild, they do not typically even have defined cell boundaries or content. It is a separate also challenging problem to group text lines or segments, as output by an OCR system, into cells with techniques similar to the ones used in this paper. There are already many works on this problem, and we wished to keep the focus on a more specific problem for which solutions are not already available. As a result, we leveraged the already publicly available IBM FinTabNet dataset (Zheng et al., 2021), which has more than 100K real tables from SEC filings already annotated with cell, row, and column boundaries, to create out datasets. In a realistic deployment scenario, our model would occupy a place in a multi-stage pipeline, downstream from the systems performing OCR, table recognition, and table and cell boundary detection.

Data & Code: We will open source our data and code on our website (details suppressed for double blind review).

2 Related Work

At the highest level, we can draw a sharp distinction between the problem of fine-grained table structure considered in this work and the vast majority of table-understanding literature, which focus on;

Upstream tasks. Detection of tables in the context of a larger, scanned document, and identification of the basic table structures, namely cells, rows, and columns, usually in the form of bounding boxes. Representative works on these tasks include (Paliwal et al., 2019; Prasad et al., 2020; Zheng et al., 2021). For a comprehensive recent survey on this topic, see (Hashmi et al., 2021).

Downstream tasks. Information extraction tasks which take as input table(s) which have already been extracted into a machine-readable form. These tasks include Question answering (Yin et al., 2020; Herzig et al., 2020, 2021; Zayats et al., 2021), Fact retrieval (Dong and Smith, 2021), Table to text generation (Wang et al., 2020; Parikh et al., 2020). For a comprehensive survey of recent advances on this topic, see (Pujara et al., 2021).

For the remainder of this section, we will focus on explaining the much smaller number existing works which address table structure more fine-grained than simple row/column membership and how our approach differs from or enhances them.

Heuristic-based approach. One of the earliest works on fine-grained table structure is (Chen et al., 2017). This work develops a heuristic approach, based on hand-crafted features, for elucidating semantic relationships between row headers only. More recently, (Wang et al., 2021) develops neural representations of tables with complex structure for use in downstream tasks, but relies on heuristics to elucidate the hierarchical structure itself. In contrast, our approach, being data-driven and based on end-to-end training of neural networks, is designed to classify cell types and identifying hierarchical relationships between row headers, column headers, captions, and content blocks, without using any heuristics.

Hybrid approach. The approach taken in (Sun et al., 2021) to automatically reconstructing table structure involves both the use of pre-trained networks to embed cells and rules enforced via PSL that express the authors' hypothesis of the relationships that are likely to occur among cells and blocks with different (fine-grained) semantic content types. (Chi et al., 2019) also use hand crafted features for representing table cells into vertices and edges, then use a graph neural network for predicting the horizontal and vertical relations between cells. In contrast, we do not incorporate any such explicit rules or hand crafted features, but fine-tune all weights of LayoutLM, a network which is pre-trained on a large and diverse document corpus, harnessing transfer learning to automatically learn a general predictive model from the data.

Neural Approaches. While there are a few completely neural approaches to extracting the structure of complex tables from images, most, such as (Xue et al., 2019) and (Qiao et al., 2021) rely on visual features alone. An exception is (Zhang et al., 2021), which relies on both visual and textual features, but still differs in two important ways from our approach. First, in contrast to LayoutLM, their model has pre-trained, separate visual and textual embeddings of the cells which are melded in a somewhat ad-hoc way into a unified cell embedding. Second, since they interpret the problem of table hierarchy elucidation as one of drawing the cell boundaries correctly, they put a limitation on the sorts of relations their system can predict. For example, multi-level (beyond 2 layer) header hierarchies, as well as parent-child relationships between cells which do not border one another (as is frequently seen in the case of row hierarchies) cannot be elucidated by their system, whereas our framework is able to handle such cases naturally.

3 Dataset Creation

In this section we discuss details of IBM Fintabnet, followed by our annotation methodology and neural model.

3.1 IBM Fintabnet

IBM FinTabNet (Zheng et al., 2021) contains 112,887 tables spread over 89,646 pages of S&P500 companies earning reports. For each table dataset provides table bounding boxes, cell bounding boxes, and the textual content of the cell. The dataset was created by passing images of PDF documents through a series of object detection and image classification neural networks. IBM's technique for producing FinTabNet achieves 99.31 F1 scores of ICDAR2013 (Göbel et al., 2013) table recognition benchmark, making it the sate-of-the-art technique at the time of writing this paper.

3.2 Data Annotation

Annotators labeled both the cell types and the parent-children relationship present among the column header cells, helping us capture the hierarchy structure of the table. Allen AI open-source tool PAWLS (Neumann et al., 2021) was used to perform annotations.

Our annotations were performed in two rounds. For the first round, we randomly sampled 500 samples from the base dataset. After a round of model training (details in Section 3.3) on the initial samples, we used the trained model to help select the samples for labeling in the second round. The aim of this is to selectively sample and annotate tables which are more likely to have complex column hierarchy than tables randomly chosen from FinTabNet. Finally, combining two rounds of annotation we manually annotate 887 tables. Table 1 provides label level information about our annotated dataset.

3.3 Modeling and Soft Labels

We tried three baseline methods: 1) Heuristics 2) BERT(Devlin et al., 2018) and 3) LayoutLM(Xu et al., 2020). For heuristic model, we detected the largest consecutive group of numeric values and

Table 1: Details of manually annotated dataset.

# of samples	525
# of tables	887
Mean Cell count	36.5±41.6
Mean Column header count	4.7±3.1
Mean Row header count	7.4±9.4
Mean Content count	23.6±32.4
# of tables with hierarchy	458

Table 2: Baseline Results

	Accuracy	Macro F1
	Cell label prediction	
Heuristic	48.27 ± 31	35.77 ± 27
BERT	95.84 ± 9	90.17 ± 17
LayoutLM	97.75 ± 9	95.08 ± 14
	Cell relation prediction	
Heuristic	73.21 ± 28	66.06 ± 32
BERT	77.27 ± 31	67.99 ± 35
LayoutLM	77.95 ± 30	69.70 ± 35

marked those as content cells. Cells above this group are labeled as column headers, and on the left of this group are marked as row headers.

In case of neural models, we model the cell label prediction task as a token classification task (e,g, Named Entity Recognition). Input is passed to the model at the token level, and cell embeddings are created by performing average pooling over all the tokens of a cell. Column hierarchy prediction is modeled as a binary classification task. For all possible column header pairs, cell embeddings are concatenated and passed onto a non-linear classifier. All models are trained end-to-end.[1]

LayoutLM achieves an F1 score of 95.08 and 69.7 on cell label prediction and relation prediction respectively. Table 2 shows the full results for both the tasks. Finally, the model is used to create soft labels for entire IBM FinTabNet dataset.

4 Discussion

Practical importance of work As mentioned above, most work on information extraction from tables does little to take account of hierarchical relationships between header cells. While tables with such complex structure are relatively rare in public datasets, the situation is quite different for proprietary datasets. For example, of the hundreds of different counterparties (external funds) submitting *capital statements* to one department of a large financial institution, it was found (by manual inspection undertaken by the authors) that roughly 30% regularly present financial results in tables have complex hierarchical header structure. Giving the information extraction models access to the finer aspects of the table structure may lead to more accurate and interpretable predictions, and even enable the business user to define certain extractions using simple business rules operating on the output of our model.

Difficulty of task Although at first glance it may seem that the problem can be adequately addressed through simple heuristics, the heuristics we tried were significantly outperformed by the best LayoutLM based model on our data (see results Table). Further, even the strong LayoutLM baseline have a high standard deviation leaveing room for improvement, particularly on the column-header relation-identficiation task.

FAIR: The community can find our dataset on our website[2] and will include all necessary metadata to ensure machine *Findable*. To ensure *Accescablity*, our data will be available using standard and universal protocols. Finally, to ensure *Interoperablity* and *Reusablity* our data will be formatted in standard formats like JSON, and we will provide detailed documentation.

5 Conclusion and Future Work

By releasing a large public dataset (by augmenting the annotations in FinTabNet with further fine-grained structure), and demonstrating performance of some strong baselines, we hope to stimulate work in the community on this still largely unsolved problem. Among the next steps to be taken are further expanding the annotations by increasing the number and diversity of tables annotated manually, and also annotating the row hierarchy structure, and caption-to-content block relationships. Further, we plan to use the structure annotations produced by our model within a pipeline, and show their utility in improving the performance of downstream extractions. Additionally, we will use the observations above concerning failure modes of the current models to motivate improvements in the structure-resolution models in an effort to improve on the LayoutLM-based baseline.

[1] Models are validated on a randomly sampled test set of 20% size and are implemented in Keras and huggingface. Each model is trained with a learning rate of $3e^{-5}$, early stopping (patience 5) on a Nvidia RTX A6000 GPU.

[2] details suppressed for blind review

References

Xilun Chen, Laura Chiticariu, Marina Danilevsky, Alexandre Evfimievski, and Prithviraj Sen. 2017. A rectangle mining method for understanding the semantics of financial tables. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 268–273. IEEE.

Zewen Chi, Heyan Huang, Heng-Da Xu, Houjin Yu, Wanxuan Yin, and Xian-Ling Mao. 2019. Complicated table structure recognition.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Rui Dong and David A Smith. 2021. Structural encoding and pre-training matter: Adapting bert for table-based fact verification. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2366–2375.

Max Göbel, Tamir Hassan, Ermelinda Oro, and Giorgio Orsi. 2013. Icdar 2013 table competition. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1449–1453. IEEE.

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*.

Jonathan Herzig, Thomas Müller, Syrine Krichene, and Julian Martin Eisenschlos. 2021. Open domain question answering over tables via dense retrieval. *arXiv preprint arXiv:2103.12011*.

Jonathan Herzig, Paweł Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Martin Eisenschlos. 2020. Tapas: Weakly supervised table parsing via pre-training. *arXiv preprint arXiv:2004.02349*.

Mark Neumann, Zejiang Shen, and Sam Skjonsberg. 2021. Pawls: Pdf annotation with labels and structure.

Shubham Singh Paliwal, D Vishwanath, 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 *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 128–133. IEEE.

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 *Proceedings of EMNLP*.

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.

Jay Pujara, Pedro Szekely, Huan Sun, and Muhao Chen. 2021. From tables to knowledge: Recent advances in table understanding. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 4060–4061.

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. *Lecture Notes in Computer Science*, page 99–114.

Kexuan Sun, Harsha Rayudu, and Jay Pujara. 2021. A hybrid probabilistic approach for table understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(5):4366–4374.

Zhenyi Wang, Xiaoyang Wang, Bang An, Dong Yu, and Changyou Chen. 2020. Towards faithful neural table-to-text generation with content-matching constraints. *arXiv preprint arXiv:2005.00969*.

Zhiruo Wang, Haoyu Dong, Ran Jia, Jia Li, Zhiyi Fu, Shi Han, and Dongmei Zhang. 2021. Tuta. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery Data Mining*.

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 *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1192–1200.

Wenyuan Xue, Qingyong Li, and Dacheng Tao. 2019. Res2tim: reconstruct syntactic structures from table images. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 749–755. IEEE.

Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. TaBERT: Pretraining for joint understanding of textual and tabular data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8413–8426, Online. Association for Computational Linguistics.

Vicky Zayats, Kristina Toutanova, and Mari Ostendorf. 2021. Representations for question answering from documents with tables and text. *arXiv preprint arXiv:2101.10573*.

Zhenrong Zhang, Jianshu Zhang, and Jun Du. 2021. Split, embed and merge: An accurate table structure recognizer. *arXiv preprint arXiv:2107.05214*.

Xinyi Zheng, Doug Burdick, Lucian Popa, Peter Zhong, and Nancy Xin Ru Wang. 2021. 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).